İZMİR EKONOMİ ÜNİVERSİTESİ

CE 477 - Data Science

2022 - 2023 Fall

Assignment 3 - Final Report

Eray Emekli - 20190614046

Mustafa Alan - 20180601003

Gizem Kılıç - 20190601030

Buse Özel - 20180601032

Data Set Link

https://www.kaggle.com/datasets/rodsaldanha/arketing-campaign?select=marketing_campaign.csv

Github Link

https://github.com/mustafaalann/CE477-DataScience-MarketingCampaign

CONTENTS

1. Introduction

We live in a world where data flows in huge amounts and diversity. When we surf the Internet, buy coffee with our credit card, or send an email to someone, we always leave a digital footprint behind us. The data knows too much about us. For example; Online shopping sites know which ads to show you in which areas, your shopping cards know your favourite brands, and by using this information, you can make your shopping experience more personal.

Data science aims to extract meaningful data from past, present and future data and plays an important role in almost all aspects of business operations and strategies. For example, data science provides us with insights about customers to help companies create stronger marketing campaigns and targeted advertising to increase product sales. In our project, which we call Marketing Campaign, we aim to maximise the profit of the company's next marketing campaign by using the data of our customers in this direction.

Therefore, our expectation is increased efficiency and reduced costs.

Identifying your data fields here will be an important part of your data strategy. In our project, each customer has a customer Id. And about them; we record and keep information such as the year they were born, their education level, marital status, annual household income, number of minors in their household, number of teenagers, dates of registration with the company, number of days since last purchase, and amounts spent on wine products in the last 2 years.

As it is known, in many areas of our lives; Data science is used in politics, health, travel, troubleshooting, agriculture, banking and many more. Our main goal in our project is to apply the information we learned in the lesson by using this data we keep about our customers.

2. Data Set Description

➢ 2.1 Description of the Data Set:

By increasing responses or lowering expenses, a response model can significantly improve a marketing campaign's efficiency. Predicting who will respond to a product or service offer is the idea.

➢ 2.2 Metadata:

- Collaborator: Rodolfo Saldanha
- Collection Methodology: Business analytics using SAS Enterprise Guide and SAS Enterprise Miner.
- Licence: O. Parr-Rud. Business Analytics Using SAS Enterprise Guide and SAS Enterprise Miner. SAS Institute, 2014.

➢ 2.3 Size of the Data:

There are 2240 instances and 29 attributes.

➢ 2.4 The Number of Missing Values in Each Attribute:

The problem of missing value is quite common in many real-life datasets. Missing value can bias the results of the machine learning models and/or reduce the accuracy of the model.

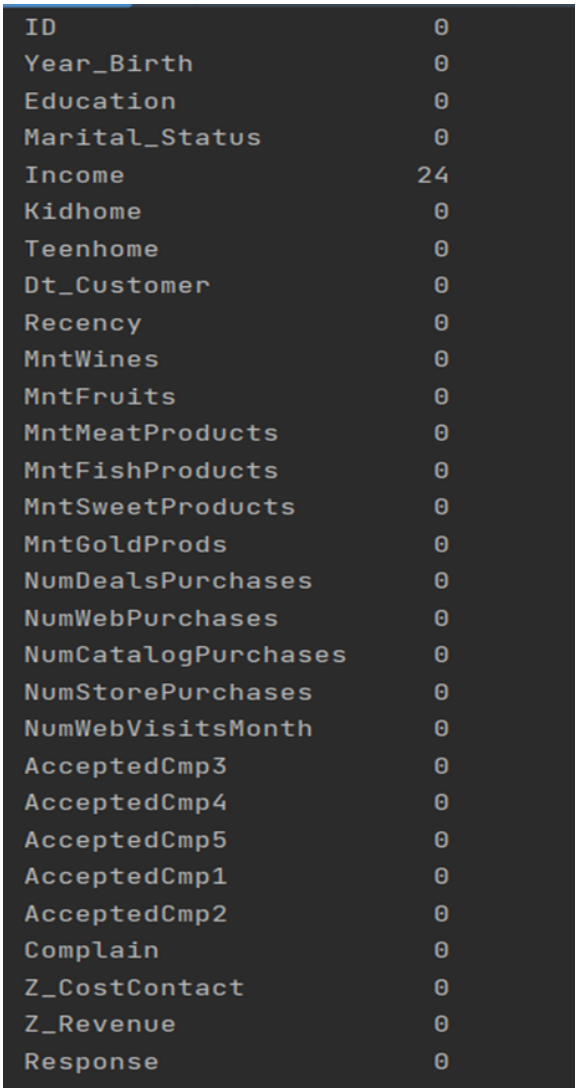


Figure #1 : The Number of Missing Values

Missing data are defined as values or data that are not stored (or not available) for some variable in the given dataset.

Figure #1 shows the total number of missing values for each attribute.

➢ 2.5 Box Plots of Numeric Attributes:

The boxplots are written using the matplotlib. The box part is the IQR of the attribute and the arms that spread the upper and lower part of the box are respectively maximum and minimum. The values outside these are the outliers which we'll find later on.

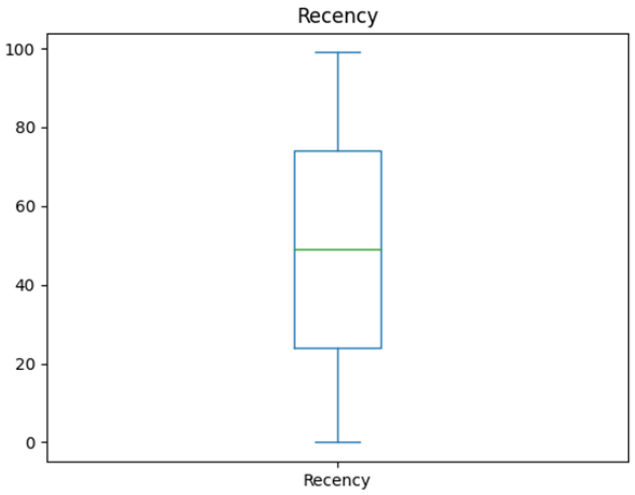


Figure #2 : The Boxplot of Attribute Recency

We can observe the boxplot of Recency in the **Figure #2**. As we can see this attribute has no outliers and its median, which is the line that divides the box, is nearly centred.

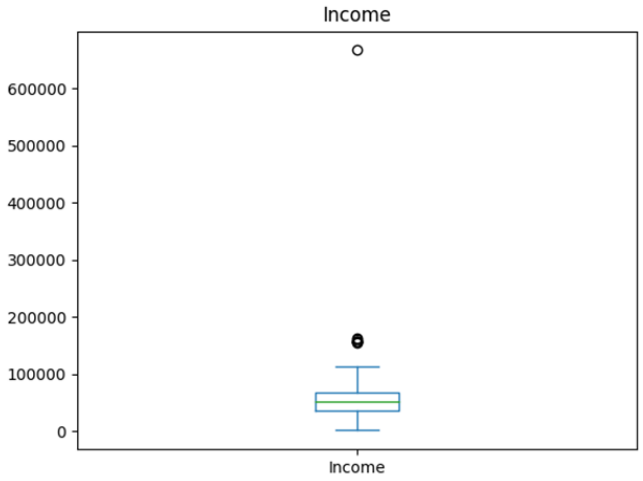


Figure #3 : The Boxplot of Attribute Income

As we observe from **Figure#3,** we have several outliers between values 200000 and 100000. Other than these outliers we have an outlier that's value being more than 600000. Income column is the only attribute that has missing values in our project and because of that to find boxplot, we used ".dropna()".

The outliers of attribute Income;

[153924.0, 156924.0, 157146.0, 157243.0, 157733.0, 160803.0, 162397.0, 666666.0]

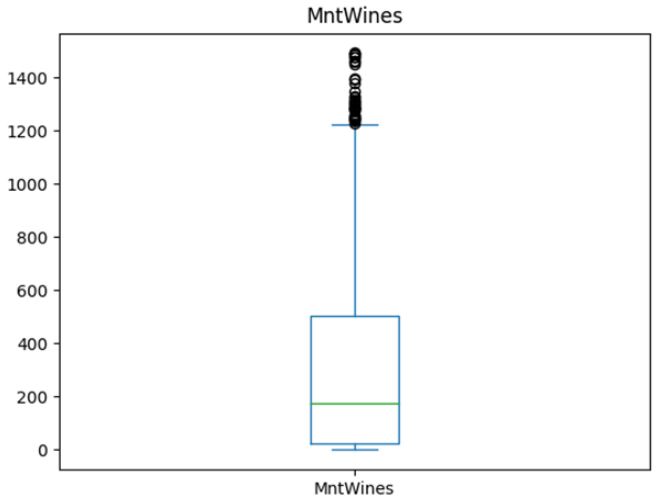


Figure #4 : The Boxplot of Attribute MntWines

In **Figure #4,** we can observe the upper fence spreading out more than the lower fence. This attribute has 35 outliers all outside of the upper fence. The median in this example is more on the lower side.

Outliers of MntWines attribute:

[1230, 1239, 1241, 1245, 1248, 1252, 1253, 1259, 1276, 1279, 1285, 1285, 1285, 1288, 1296, 1298, 1302, 1308, 1311, 1315, 1324, 1332, 1349, 1379, 1394, 1396, 1449, 1459, 1462, 1478, 1478, 1486, 1492, 1492, 1493]

➢ 2.6 Histograms for Nominal Attributes:

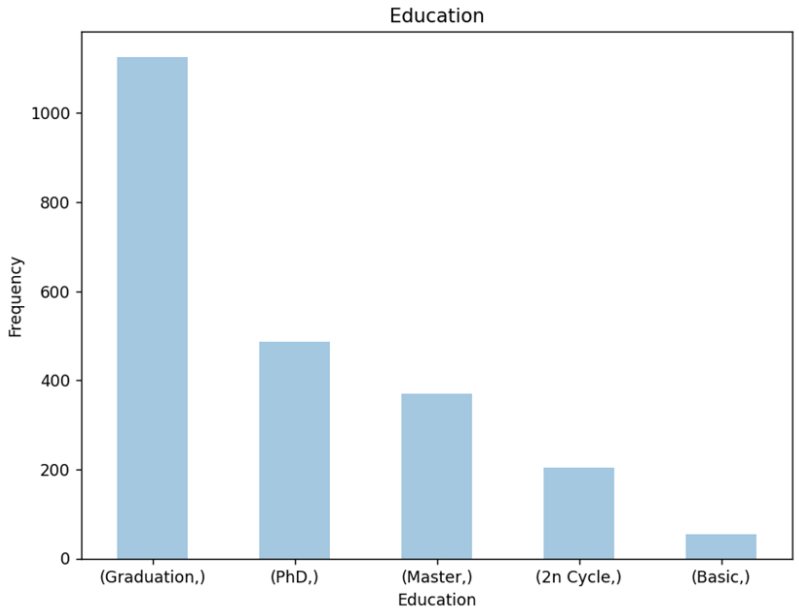


Figure #5 : Histogram of the Attribute Education

We can see the number of people who has the particular education status on the **Figure #5**. We can see that the "Graduation" type is the most common and "Basic" is the least common one.

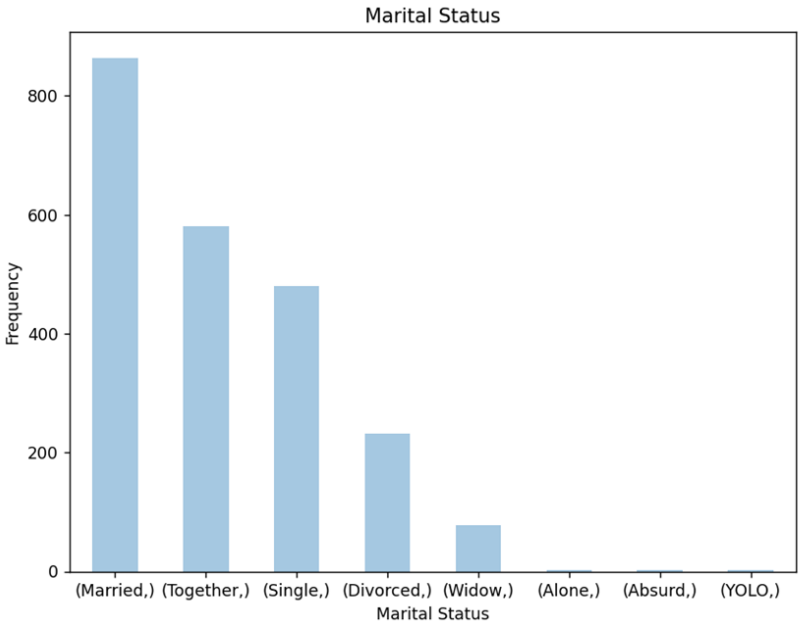


Figure #6 : Histogram of the Attribute Marital Status

We can see on the **Figure #6** that most of the people's Marital Status is Married and there are some inputs that do not make sense and might be considered as NaN.

➢ 2.7 Scatter Plots of Attributes:

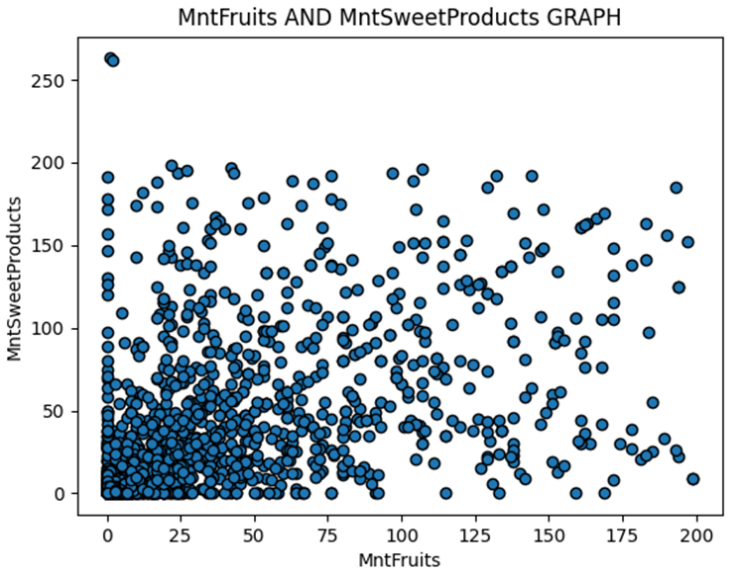


Figure #7 : Scatter Plot of the Attributes MntFruits and MntSweetProducts

As we can see from **Figure #7**, the middle part is more crowded, and there is almost a line visible. Although it is not very obvious, we can assume that there is a relationship between the amount of time spent on sweet products and fruits.

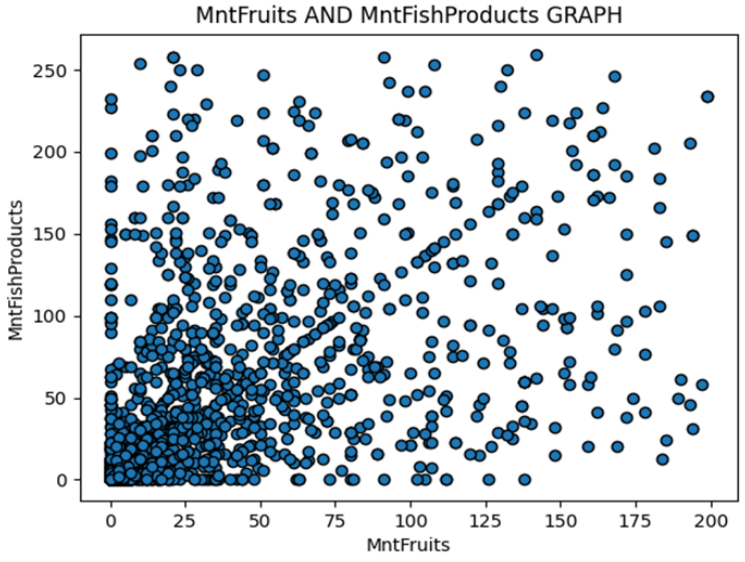


Figure #8 : Scatter Plot of the Attributes MntFruits and MntFishProducts

As we can see from **Figure #8**, again, there is almost a line visible in the middle. Although it is not very obvious, we can assume that there is a relationship between the amount of time spent on fish products and fruits.

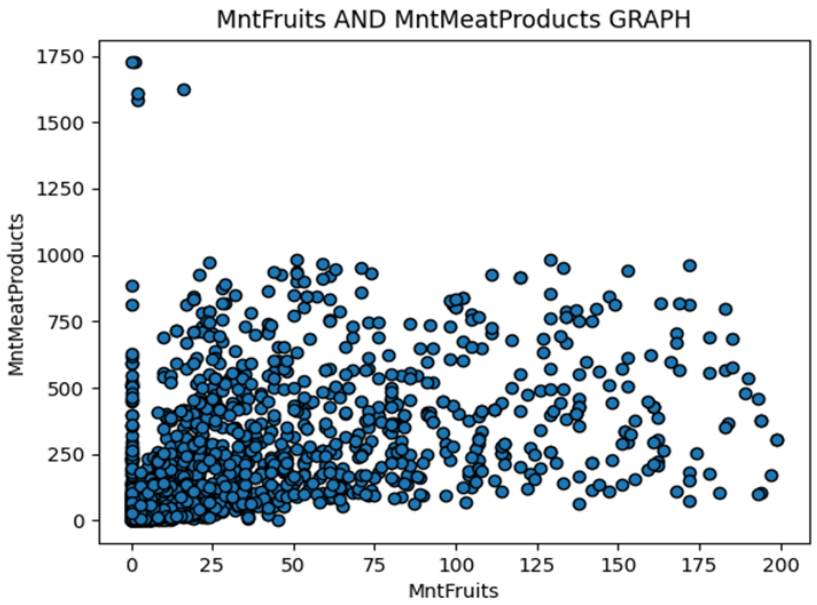


Figure #9 : Scatter Plot of the Attributes MntFruits and MntMeatProducts

In **Figure #9,** we see the relationship between time spent on meat products and fruits.

3. Preprocessing

➢ 3.1 The Correlation Matrix of the Attributes:

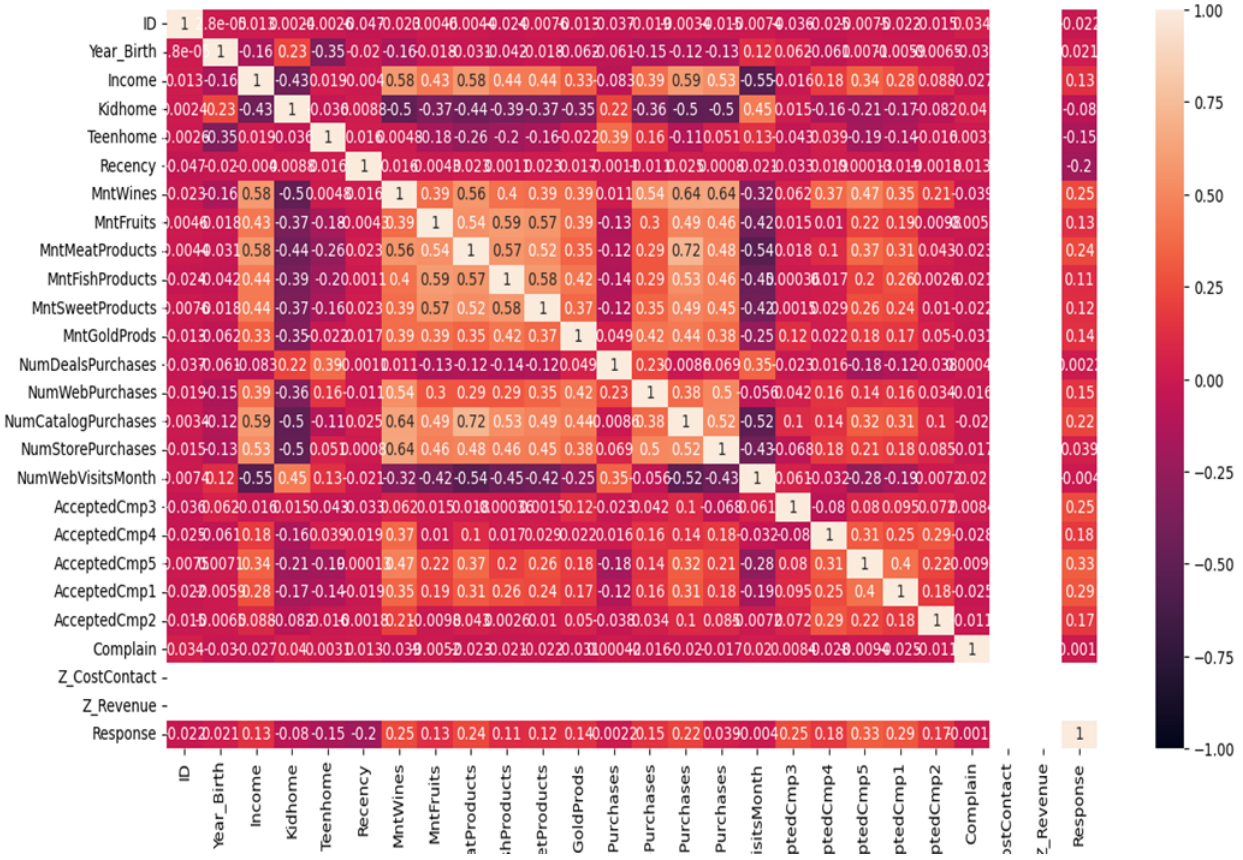


Figure #10 : The Correlation Matrix Table

The Correlation Matrix is a matrix table that shows the correlation levels of the scale expressions with each other using the correlation coefficient. Pandas' correlation matrix function corr() uses "Pearson" analysis by default. Therefore, non-numeric data types will be excluded from this correlation.

As can be seen in **Figure #10**, the matrix table is divided into 2 diagonally in the middle. In this graph, dark colours like black represent negative correlation, while light colours like white represent positive correlation.

➢ 3.2 Attribute Selection:

Attribute selection is an important issue in Machine Learning that affects performance a lot. There are many variable selection methods for this. However, we used the Correlation method, which is used when the input and output variables are numeric in the previous sub-title.

In this method, the condition that the variables are linear and normal distribution is sought. The result of the test gives the value and direction of the relationship between the two variables. Accordingly, values close to 0 or 0 indicate no relationship, a value close to 1 indicates a positive relationship, and a value close to -1 indicates an inverse/negative relationship.

As seen in **Figure #10**, MntMeatProducts and NumCatalogPurchases have strong positive correlations with 0.72, and NumWebVisitsMonth and Income have strong negative correlations with -0.55.

According to the attribute selection, if there is a strong correlation between the variables, they should be grouped under the same factor.

➢ 3.3 Discretization on Some of the Numeric Attributes:

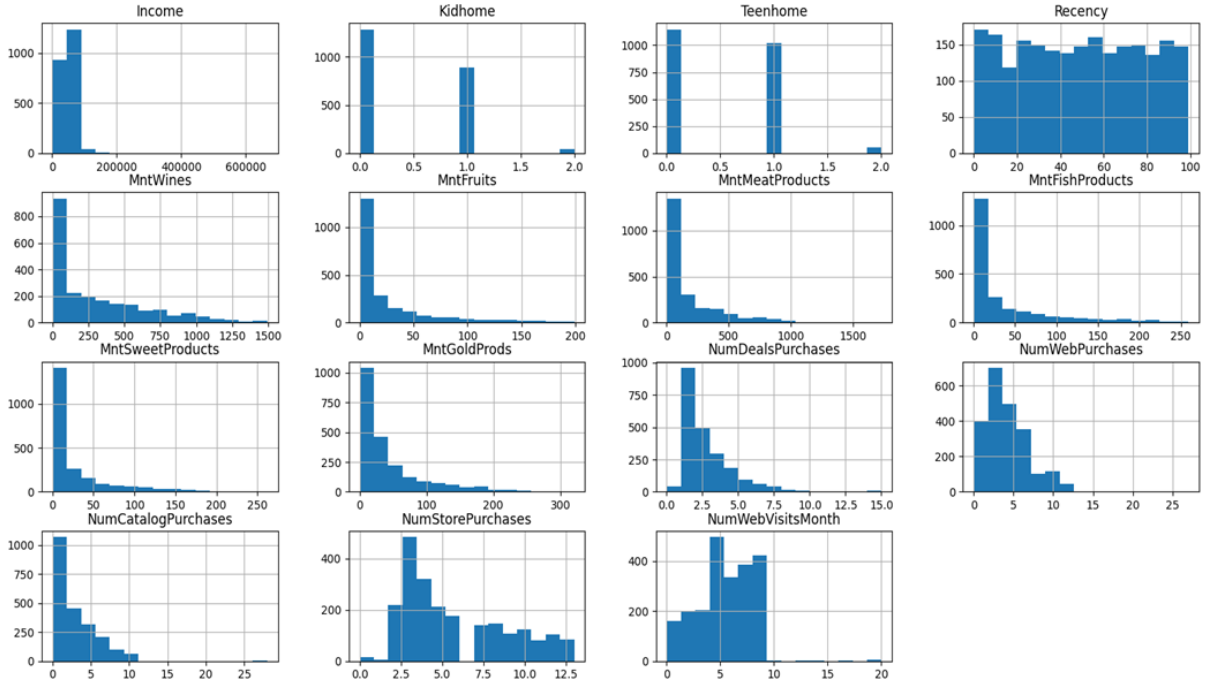


Figure #11 : Discretization on Numeric Attributes

Numerical input variables may have non-standard distribution. It can be caused by outliers in the data/multi-modal distributions/highly exponential distributions or more. Today's machine learning algorithms perform better on the numerical input variables which have a standard probability distribution. In order to achieve that; data scientists and coders use data discretization. Discretization provides an automatic way to change a numeric input variable to have a different data distribution so they can be used as input to a predictive model. So values of those variables are grouped together into discrete bins and those bins are assigned a unique integer like ordinal relationship between the bins which is preserved.

There are different methods for grouping the values into k discrete bins; in order to create **Figure #11**, Uniform technique is used.Which means;the "strategy" selected as"uniform" and since uniform is flexible; so n_bins do not have to be less than the number of observations.

We can see that in **Figure#11**, the histograms match with the shape of the raw dataset.

➢ 3.4 Normalization and Standardization on Some of the Numeric Attributes:

Before Normalization

	Income	MntWines
0	58138.0	635
1	46344.0	11
2	71613.0	426
3	26646.0	11
4	58293.0	173
...	...	...
2235	61223.0	709
2236	64014.0	406
2237	56981.0	908
2238	69245.0	428
2239	52869.0	84

After Normalization

	Income	MntWines
0	0.999940	0.010922
1	1.000000	0.000237
2	0.999982	0.005949
3	1.000000	0.000413
4	0.999996	0.002968
...	...	...
2235	0.999933	0.011580
2236	0.999980	0.006342
2237	0.999873	0.015933
2238	0.999981	0.006181
2239	0.999999	0.001589

Figure #12 : Normalization of the Attributes Income and MntWines

We removed the instances with Nan income values from the dataframe then we rescaled the Income and MntWines values. As you can see from the **Figure #12**, after normalization now we can see that the difference between two attributes became less and we can make analyses on them easier.

Before Standardization

	Income	MntWines
0	58138.0	635
1	46344.0	11
2	71613.0	426
3	26646.0	11
4	58293.0	173
...	...	...
2235	61223.0	709
2236	64014.0	406
2237	56981.0	908
2238	69245.0	428
2239	52869.0	84

After Standardization

	Income	MntWines
0	0.234063	0.978226
1	-0.234559	-0.872024
2	0.769478	0.358511
3	-1.017239	-0.872024
4	0.240221	-0.391671
...	...	...
2235	0.356642	1.197646
2236	0.467539	0.299208
2237	0.188091	1.787710
2238	0.675388	0.364441
2239	0.024705	-0.655568

Figure #13 : Standardization of the Attributes Income and MntWines

We removed the instances with Nan Income or MmntWines values from the dataframe then we rescaled the Income and MntWines values. As you can see from the **Figure #13**, after

standardization now we can see that the difference between two attributes became less and we can make analyses on them easier.

➢ 3.5 One-hot Encoding to One of the Categorical Attributes:

Applying one hot encoding to an Education attribute makes sense because there are only 5 types of educational states, which are “Graduation”, “PhD”, “Master”, “2n Cycle” and “Basic”.

----------ONE HOT ENCODED(EDUCATION)---------

	2n Cycle	Basic	Graduation	Master	PhD
0	0.0	0.0	1.0	0.0	0.0
1	0.0	0.0	1.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0
3	0.0	0.0	1.0	0.0	0.0
4	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...
2235	0.0	0.0	1.0	0.0	0.0
2236	0.0	0.0	0.0	0.0	1.0
2237	0.0	0.0	1.0	0.0	0.0
2238	0.0	0.0	0.0	1.0	0.0
2239	0.0	0.0	0.0	0.0	1.0

Figure #14 : One-Hot Encoding of the Attribute Education

As we can see from **Figure #14**, there are new columns created for each unique value of a specific feature ‘Education’.

➢ 3.6 PCA, and Visualize the Data in 2 Dimensions:

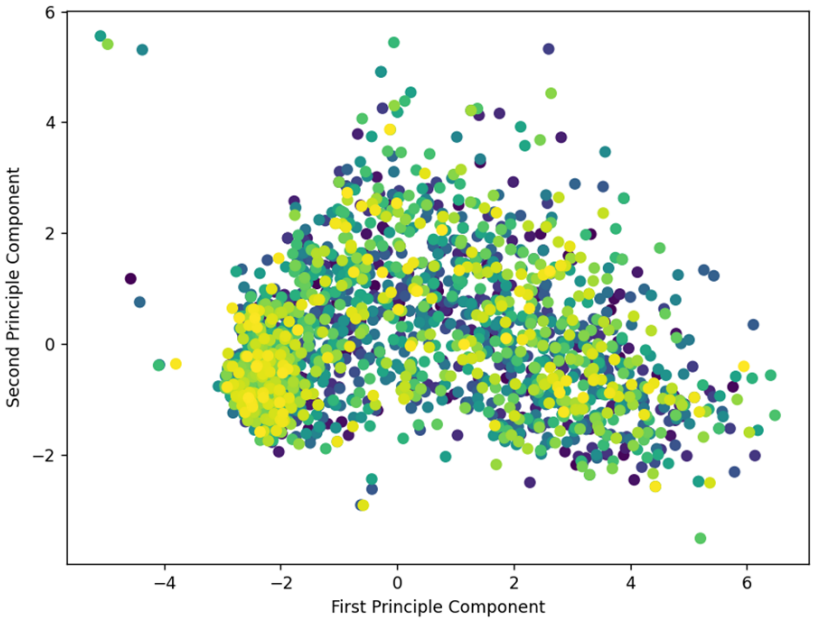


Figure #15 : PCA and 2D Visualization of Data

As we can see from the **Figure #15,** we visualized all the numeric data in 2D by using the PCA technique.

➢ 3.7 Impute for Missing Values:

There are different ways of imputing the missing values. Replace with Average is the most common method of imputing missing values for numeric columns. In the 2.4 section, we found the number of missing values in our data. We used the 'fillna' method to assign the Income column with missing values along with the average of the corresponding column values. The new results are seen in **Figure #16**.

Figure #16 : The Number of Missing Values After Imputing

➢ 3.8 Outlier Detection:

To find outliers first of all, we found IQR. To find it we used the selected attribute's third quartile minus the second quartile. Then with this value we found the upper and lower fences. With another function we go through the column and select the values which are lower than lower bound or higher than upper bound then put the values to the array to return them.

```
IQR of attribute Year_Birth:
18.0
Lower fence of Year_Birth:
1932.0
Upper fence of Year_Birth:
2004.0
Number of outliers:
3
Outliers:
[1893, 1899, 1900]
```

Figure #17 : The Outliers of Attribute Year_Birth

As we can see from the **Figure #17** there are 3 outliers of Year_Birth. This attribute's IQR is 18.0. With this IQR we found lower and upper fences, respectively 1932 and 2004. After getting through the column of attribute Year_Birth we found that three values are outliers. These outliers are also all lower than the lower fence value and none of them are from the outside of the upper fence.

```
IQR of attribute MntWines:
480.5
Lower fence of MntWines:
-697.0
Upper fence of MntWines:
1225.0
Number of outliers:
35
Outliers:
[1230, 1239, 1241, 1245, 1248, 1252, 1253, 1259,
```

Figure #18 : The Outliers of Attribute MntWines

Unlike the previous example in **Figure #18**, attribute MntWines' outliers are all outside of the upper fence. The issue with this is the lower fence amount since shopping for a negative amount of wine is not possible.

➢ 3.9 Data Augmentation:

Data augmentation is used to generate new data from existing ones to increase the number of samples. It is mostly used for visual data when machine learning is learning visual content. We have enough data. That is why we are not planning to use any data augmentation technique.

4. Classification

➢ 4.1 Selecting a discrete attribute as a target attribute:

We have selected our target attribute as 'Response' because the value of the attribute is 1 if there is response and 0 if there is not.

➢ 4.2 Splitting the data set into training and testing:

We have divided the dataset into 33% for testing and 66% for training. We explained it in detail in part 2.2.

➢ 4.3 Training 3 classification algorithms:

4.3.1 Decision Tree:

First of all we had to use the One-Hot-Encoding technique to make our values numeric and let the decision tree do its job. Attributes such as 'Education' and 'Marital_Status' have string values. So we replaced them with their one-hot-encoded versions. (We have showed how the result looks in the previous submission)

Secondly, we split the test into 4 different parts which are X_train, X_test, y_train and y_test. X has all the attributes apart from Response because we want to predict Response values. and y is the response part. (We used ⅓ of the data as Testing data set and ⅔ for Training data.)

Thirdly, we have used DecisionTreeClassifier to train our program.

```
--------------------------
Accuracy
0.8324324324324325
```

Figure #19: Accuracy of predictions

As you can see on **Figure#19** when we compare our predictions of X_test with y_test. Our results match %83.2. Because; the accuracy score is high, we can predict any result easily.

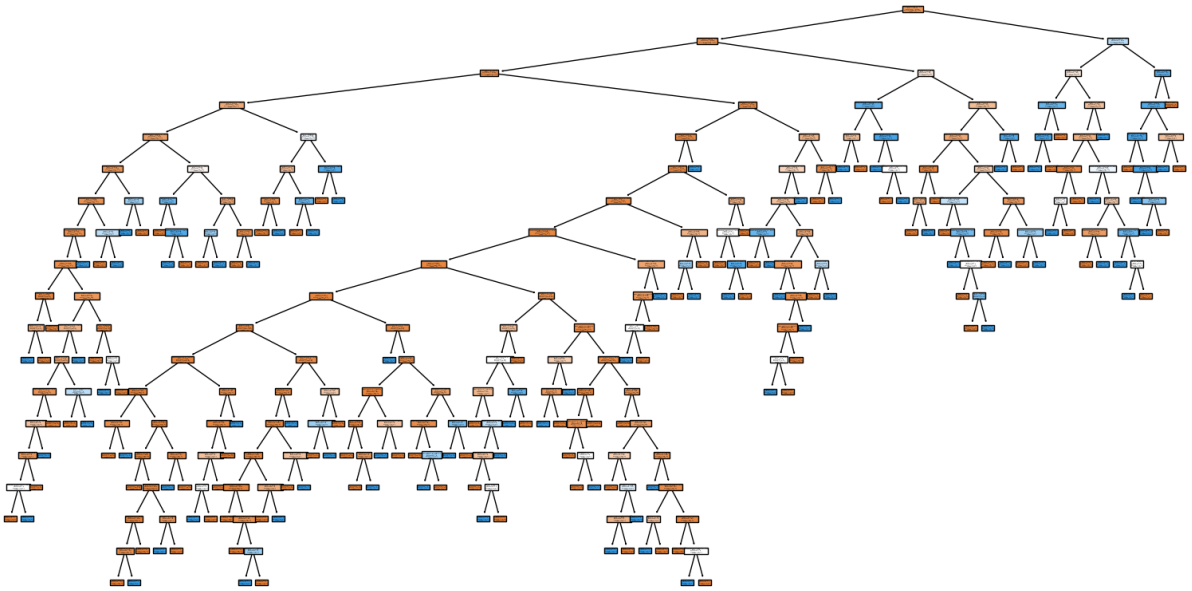


Figure #20: Decision Tree

As you can see on the **Figure #20** We have a huge decision tree due to our attribute numbers.

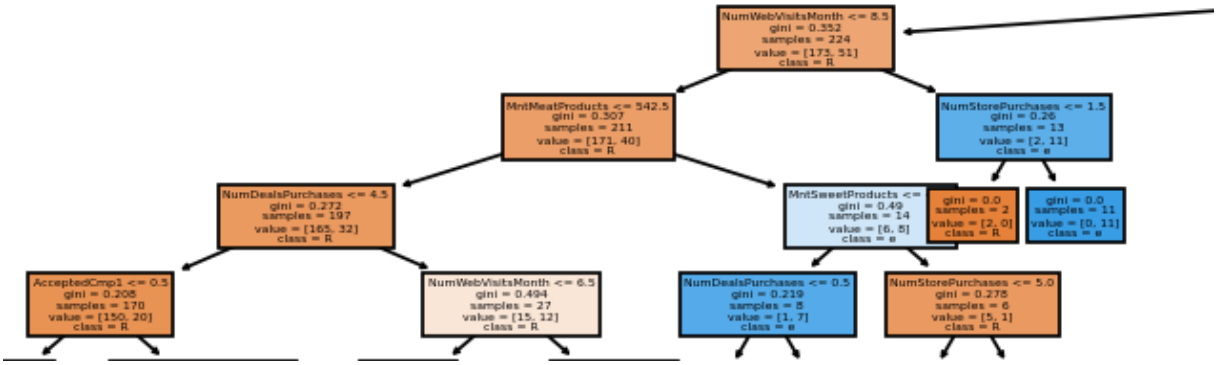


Figure #21: Decision Tree

You can see a part of the Decision Tree on the **Figure #21.**

4.3.2 KNNeighborClassifier:

```
------------------------------
KNN - Score
0.7756756756756756
```

Figure #22: The Accuracy(Score) of KNNeighborClassifier when n_neighbors=1

We have used KNeighborClassifier with the same Train and Test data(We used ⅓ of the data as Testing data set and ⅔ for Training data.). Now we got the KNN score as 0.78 as you can see from the **Figure #22**.

```
------------------------
KNN - Score
0.8391891891891892
```

Figure #23: The Accuracy(Score) of KNNeighborClassifier when n_neighbors=3

We can see that the Accuracy is higher on the **Figure #23** which means when we change the number of neighbours which help us to predict the type of our sample to 3, the accuracy increases.(We used ⅓ of the data as Testing data set and ⅔ for Training)

4.3.3 RandomForestClassifier:

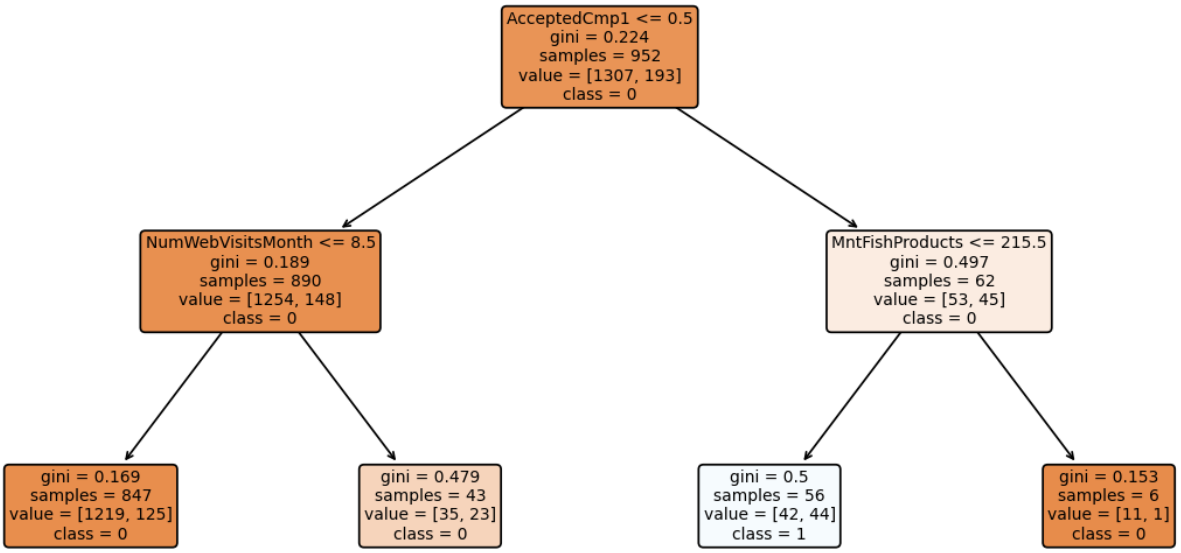


Figure #24

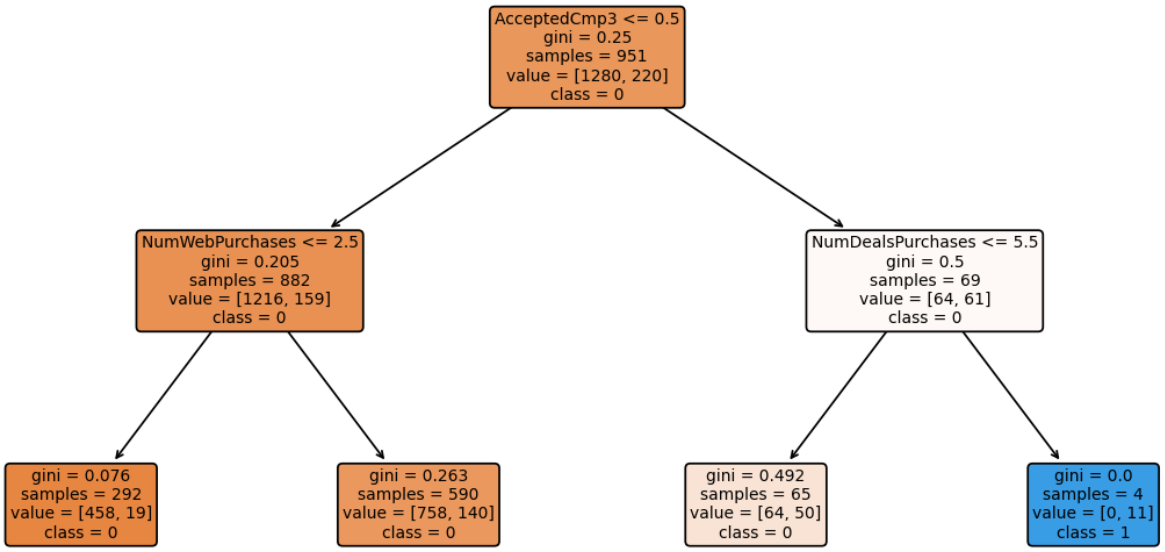


Figure #25

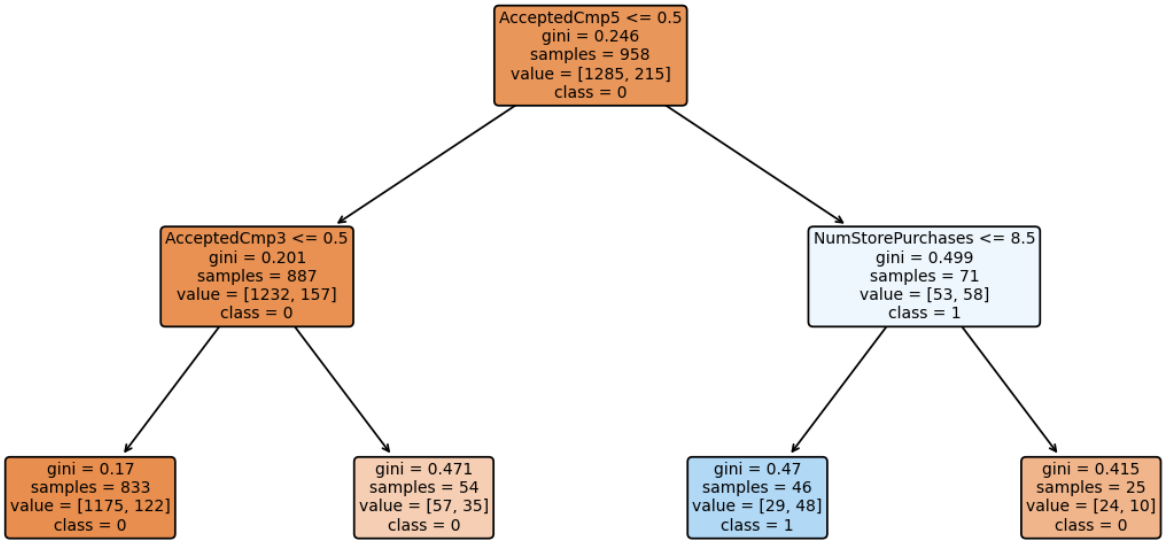


Figure #26

As you can see above. **Figure #24, Figure #25** and **Figure #26** are the 3 decision trees which are shown from a RandomForestClassifier algorithm. With just three estimators, it's clear how scaling up gives a rich, diverse representation of the knowledge that can be successfully assembled into a highly-accurate model.(We used ⅓ of the data as Testing data set and ⅔ for Training data.)

Note: *The more trees there are in the forest, the more varied the model can be. There is a point of diminishing returns, however, because with many trees fit on a random subset of features, there will be a fair amount of similar trees in the ensemble that don't offer much diversity, and which will start to have too much voting power and skew the ensemble to be overfit on the training dataset, hurting generalization to the validation set.*

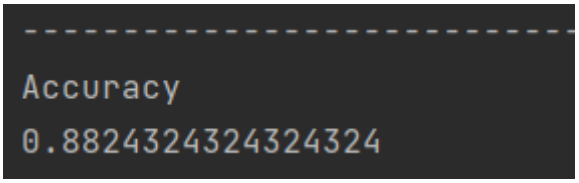


Figure #27 : Accuracy of RandomForestClassifier

We used RandomForestClassifier with the same Train and Test data. Now we have the accuracy 0.88, as you can see from the **Figure #27.**

➢ 4.4 Comparison of Algorithms :

The **DecisionTree** method gave **0.83** accuracy. **KNearestNeigbor** has max **0.84** accuracy and **RandomForest** method gives **0.88**. As we can see from the accuracy levels, RandomForestClassifier has the highest one. Which means it can predict the "Response" value in a better way.

5. Regression

➢ 5.1 Selecting a numeric attribute as a target attribute:

MntMeatProducts : The relationship between the number of meat products and other independent variables can be observed. (Used in Linear Regression Algorithm)

Teenhome : This feature has three values so using it is easier and has also a lot of positive correlation with other attributes. (Used in Multiple Linear Regression AND Logistic Regression)

➢ 5.2 Splitting the data set into training and testing:

While doing machine learning, the data is separated as training and test data, usually 80% to 20%, so that the model can be applied. While the training data is the dataset on which the model is trained, the test data is an examination of the model created in the training dataset.

Part of the dataset generated for review is used to check the adequacy of the model. This situation helps the model to test how it will perform in situations that are not part of the training, and the test result helps determine the model's performance.

We can do this using pandas DataFrames method.

```
         ID  Year_Birth   Education  ... Z_CostContact  Z_Revenue  Response
0      5524        1957  Graduation  ...             3         11         1
1      2174        1954  Graduation  ...             3         11         0
2      4141        1965  Graduation  ...             3         11         0
3      6182        1984  Graduation  ...             3         11         0
4      5324        1981         PhD  ...             3         11         0
...     ...         ...         ...  ...           ...        ...       ...
2235  10870        1967  Graduation  ...             3         11         0
2236   4001        1946         PhD  ...             3         11         0
2237   7270        1981  Graduation  ...             3         11         0
2238   8235        1956      Master  ...             3         11         0
2239   9405        1954         PhD  ...             3         11         1

[2240 rows x 29 columns]
Number of training examples: 1792
Number of testing examples: 448
```

Figure #28 : The Number of Training and Testing Examples

As seen in **Figure #28**, we have divided the dataset as 80% and 20% as training and test data.

➢ 5.3 Train at least 2 regression algorithms:

5.3.1 Linear Regression Algorithm:

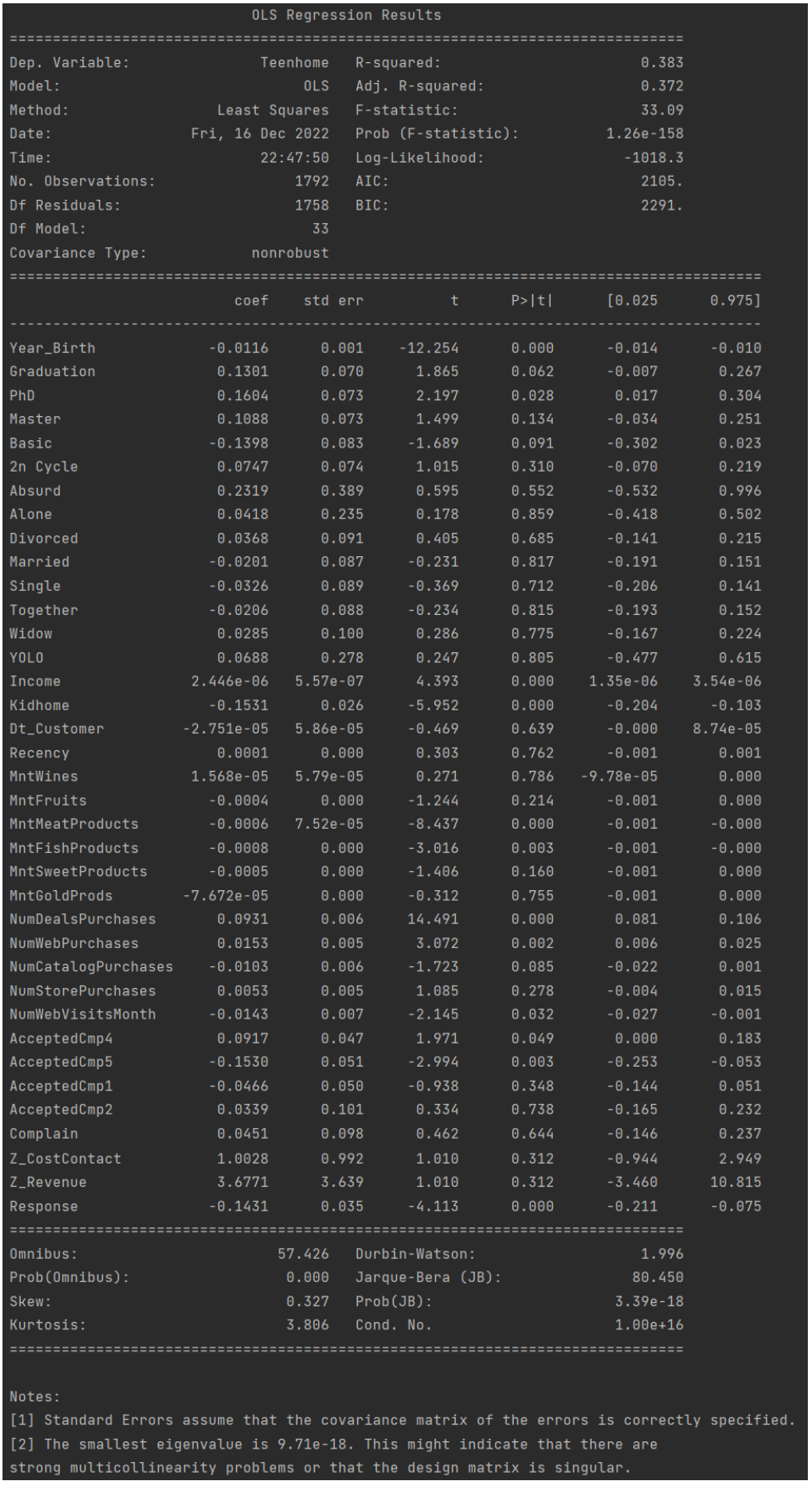
Linear Regression Algorithm, used in data science and machine learning, is a statistical method that tries to show the relationship between variables. Since we will create a line in Linear Regression, a total of 2 variables, one dependent and one independent variable, are studied. Variable selection is very important for the correct formation of the graph. For this reason, there must be a visible correlation between them.

For this process, we chose the variables MntMeatProducts and NumCatalogPurchases, which we know to have a strong correlation, based on our previous work.

Then to include our other features we used another Regression, Multiple Linear Regression. With this we took Teenhome as the learning feature and used all the other features as for training. We printed the Multiple Linear Regression Intercept and Coefficients. You can see the result in **Figure #29**. We can also observe the accuracy score which is the Score. The confidence of the accuracy is so low, this may cause many false alarms.

```
Intercept:
 43.56534948679235
Coefficients:
 [-1.16395205e-02  6.32643109e-02  9.35531035e-02  4.19672907e-02
 -2.06659789e-01  7.87508417e-03  1.90072129e-01 -2.92184164e-05
 -5.00565556e-03 -6.19323334e-02 -7.44332748e-02 -6.24003691e-02
 -1.32917185e-02  2.70204407e-02  2.44645526e-06 -1.53099837e-01
 -2.75078424e-05  1.11765197e-04  1.56820626e-05 -4.43499590e-04
 -6.34547430e-04 -8.19521703e-04 -4.83820708e-04 -7.67189189e-05
  9.31381260e-02  1.53238467e-02 -1.02901023e-02  5.34908048e-03
 -1.43311265e-02  9.16879740e-02 -1.52966101e-01 -4.65768545e-02
  3.38620562e-02  4.51145598e-02  0.00000000e+00  0.00000000e+00
 -1.43052728e-01]
Score:
 0.3809193612323284
```

Figure #29 : Multiple Linear Regression Intercept and Coefficients



OLS Regression Results

Dep. Variable:	Teenhome	R-squared:	0.383
Model:	OLS	Adj. R-squared:	0.372
Method:	Least Squares	F-statistic:	33.09
Date:	Fri, 16 Dec 2022	Prob (F-statistic):	1.26e-158
Time:	22:47:50	Log-Likelihood:	-1018.3
No. Observations:	1792	AIC:	2105.
Df Residuals:	1758	BIC:	2291.
Df Model:	33		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Year_Birth	-0.0116	0.001	-12.254	0.000	-0.014	-0.010
Graduation	0.1301	0.070	1.865	0.062	-0.007	0.267
PhD	0.1604	0.073	2.197	0.028	0.017	0.304
Master	0.1088	0.073	1.499	0.134	-0.034	0.251
Basic	-0.1398	0.083	-1.689	0.091	-0.302	0.023
2n Cycle	0.0747	0.074	1.015	0.310	-0.070	0.219
Absurd	0.2319	0.389	0.595	0.552	-0.532	0.996
Alone	0.0418	0.235	0.178	0.859	-0.418	0.502
Divorced	0.0368	0.091	0.405	0.685	-0.141	0.215
Married	-0.0201	0.087	-0.231	0.817	-0.191	0.151
Single	-0.0326	0.089	-0.369	0.712	-0.206	0.141
Together	-0.0206	0.088	-0.234	0.815	-0.193	0.152
Widow	0.0285	0.100	0.286	0.775	-0.167	0.224
YOLO	0.0688	0.278	0.247	0.805	-0.477	0.615
Income	2.446e-06	5.57e-07	4.393	0.000	1.35e-06	3.54e-06
Kidhome	-0.1531	0.026	-5.952	0.000	-0.204	-0.103
Dt_Customer	-2.751e-05	5.86e-05	-0.469	0.639	-0.000	8.74e-05
Recency	0.0001	0.000	0.303	0.762	-0.001	0.001
MntWines	1.568e-05	5.79e-05	0.271	0.786	-9.78e-05	0.000
MntFruits	-0.0004	0.000	-1.244	0.214	-0.001	0.000
MntMeatProducts	-0.0006	7.52e-05	-8.437	0.000	-0.001	-0.000
MntFishProducts	-0.0008	0.000	-3.016	0.003	-0.001	-0.000
MntSweetProducts	-0.0005	0.000	-1.406	0.160	-0.001	0.000
MntGoldProds	-7.672e-05	0.000	-0.312	0.755	-0.001	0.000
NumDealsPurchases	0.0931	0.006	14.491	0.000	0.081	0.106
NumWebPurchases	0.0153	0.005	3.072	0.002	0.006	0.025
NumCatalogPurchases	-0.0103	0.006	-1.723	0.085	-0.022	0.001
NumStorePurchases	0.0053	0.005	1.085	0.278	-0.004	0.015
NumWebVisitsMonth	-0.0143	0.007	-2.145	0.032	-0.027	-0.001
AcceptedCmp4	0.0917	0.047	1.971	0.049	0.000	0.183
AcceptedCmp5	-0.1530	0.051	-2.994	0.003	-0.253	-0.053
AcceptedCmp1	-0.0466	0.050	-0.938	0.348	-0.144	0.051
AcceptedCmp2	0.0339	0.101	0.334	0.738	-0.165	0.232
Complain	0.0451	0.098	0.462	0.644	-0.146	0.237
Z_CostContact	1.0028	0.992	1.010	0.312	-0.944	2.949
Z_Revenue	3.6771	3.639	1.010	0.312	-3.460	10.815
Response	-0.1431	0.035	-4.113	0.000	-0.211	-0.075

Omnibus:	57.426	Durbin-Watson:	1.996
Prob(Omnibus):	0.000	Jarque-Bera (JB):	80.450
Skew:	0.327	Prob(JB):	3.39e-18
Kurtosis:	3.806	Cond. No.	1.00e+16

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 9.71e-18. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Figure #30 : Multiple Linear Regression Results

In **Figure #30** We used statsmodels and printed all the features, their coefficients and standard errors.

5.3.2 Logistic Regression Algorithm:

This statistical model (also known as a logit model) is frequently used for classification and predictive analytics. Based on a set of independent variables, logistic regression calculates the probability of an event occurring, such as voting or not voting. Since the outcome is a probability, the dependent variable has a range of 0 to 1.

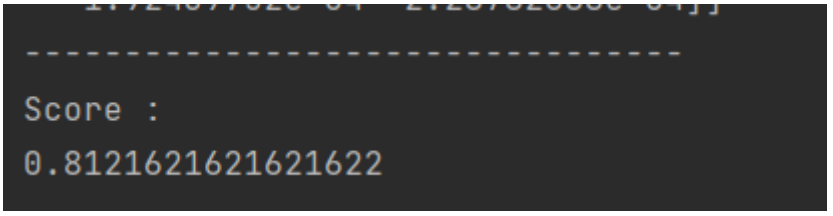


Figure #31 : Logistic Regression Intercept and Coefficients

➢ 5.4 Visualise the learned models if possible:

5.4.1 Linear Regression Algorithm Model :

As we observe the **Figure #32**, the relationship between NumCatalogPurchases and MntMeatProducts, there are clear outliers on the model. This model also proves that MntMeatPurchases has more dependencies as some of the scatters are far away from the line.

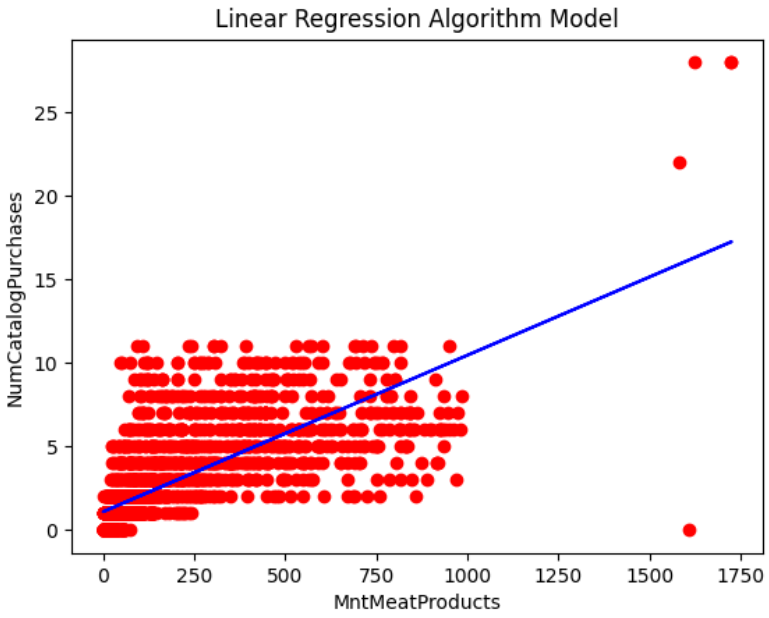


Figure #32 : Scatter Plot of Linear Regression Model of MntMeatProducts & NumCatalogPurchases

5.4.2 Logistic Regression Algorithm Model:

This model we observe in **Figure #33** depicts the actual and predicted values of Teenhome and their distribution. As we observe from the matrix number of 0 teen at home dominates the matrix. Since 2 teens at home is so minimal it gets dominated, when predicting the values it does not even appoint a value in 2. The accuracy score is high, it may not cause many false alarms.

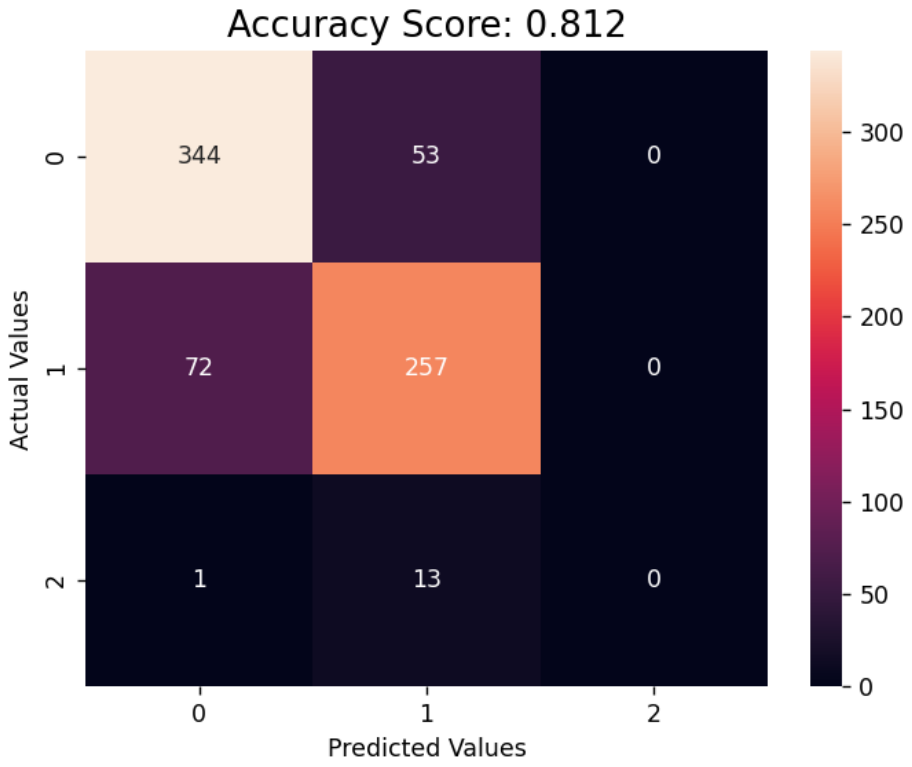


Figure #33 : Model's success metrics of Teenhome prediction /

Confusion Matrix of Logistic Regression Algorithm

➢ 5.5 Comparing the performance of the algorithms:

As we can see, the Linear Regression Algorithm's result was approximately 0.3809 and the Logistic Regression Algorithm's result was 0.812 which is much higher. According to these results we can see that predicting with the Logistic Regression Algorithm makes more sense to predict any Teenhome value.

6. Clustering

➢ 6.1 K-Means:

First we check the information about the dataframe.

```
Data columns (total 38 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   2n Cycle             2240 non-null   float64
 3   Basic                2240 non-null   float64
 4   Graduation           2240 non-null   float64
 5   Master               2240 non-null   float64
 6   PhD                  2240 non-null   float64
 7   Absurd               2240 non-null   float64
 8   Alone                2240 non-null   float64
 9   Divorced             2240 non-null   float64
 10  Married              2240 non-null   float64
 11  Single               2240 non-null   float64
 12  Together             2240 non-null   float64
 13  Widow                2240 non-null   float64
 14  YOLO                 2240 non-null   float64
 15  Kidhome              2240 non-null   int64
 16  Teenhome             2240 non-null   int64
 17  Recency              2240 non-null   int64
 18  MntWines             2240 non-null   int64
 19  MntFruits            2240 non-null   int64
 20  MntMeatProducts      2240 non-null   int64
 21  MntFishProducts      2240 non-null   int64
 22  MntSweetProducts     2240 non-null   int64
 23  MntGoldProds         2240 non-null   int64
 24  NumDealsPurchases    2240 non-null   int64
 25  NumWebPurchases      2240 non-null   int64
 26  NumCatalogPurchases  2240 non-null   int64
 27  NumStorePurchases    2240 non-null   int64
```

Figure #34 : Information of Dataframe

As you can see from the **Figure #34** There is no null value so we can continue to the K-Means algorithm to find Centroids first.

```
Centroids
[[0.99999872 0.00118829]
 [0.9999766  0.00661826]
 [0.99990664 0.01337992]]
```

Figure #35 : 3 Centroids' values

We created 3 clusters from our dataset and we can see the centroid values of them on the **Figure #35**

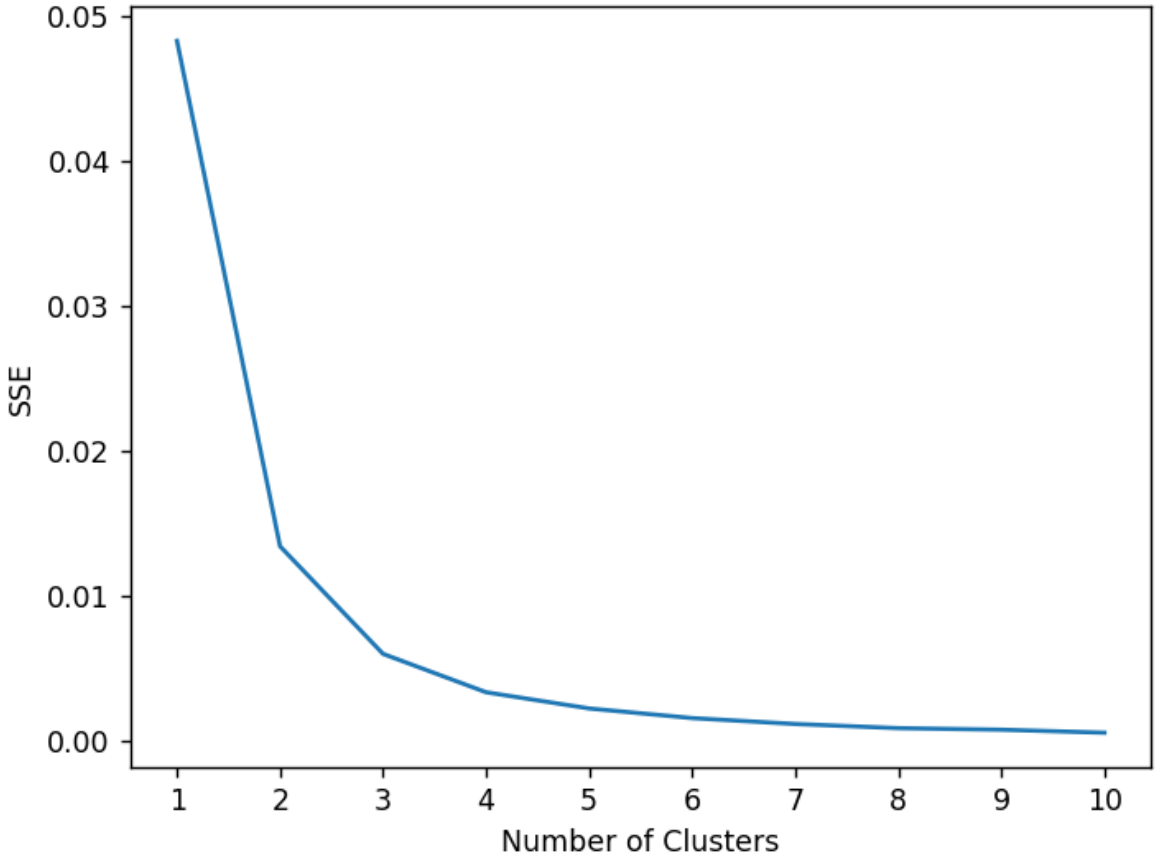


Figure #36 : Number of Clusters

"Cluster refers to a collection of data points aggregated together because of certain similarities".

We defined a target number k, which refers to our number of centroids that we needed in our dataset. The centroid is an imaginary or real location which represents the center of our cluster.

The algorithm started with a first group of randomly selected centroids as beginning points for every cluster,after this process; it performed iterative calculations in order to optimize the positions of our centroids.It continued this process until the centroids have been stabilised.

➢ 6.2 Applying hierarchical clustering algorithm:

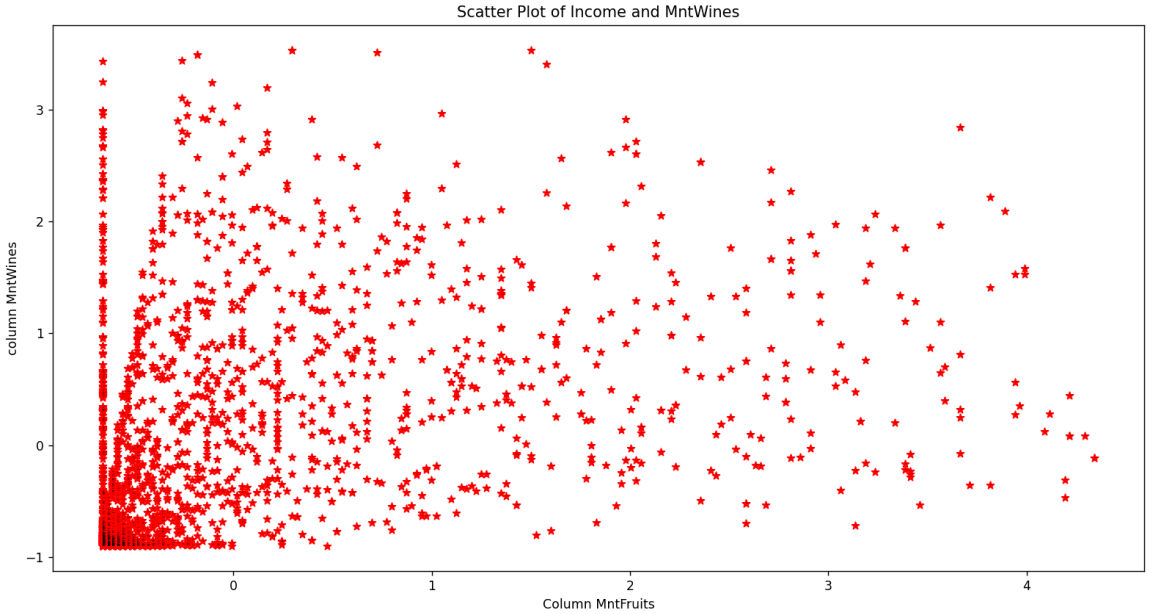


Figure #37 : Scatter Plot of Income and Wines

We can see how the data is distributed on the Scatter plot.

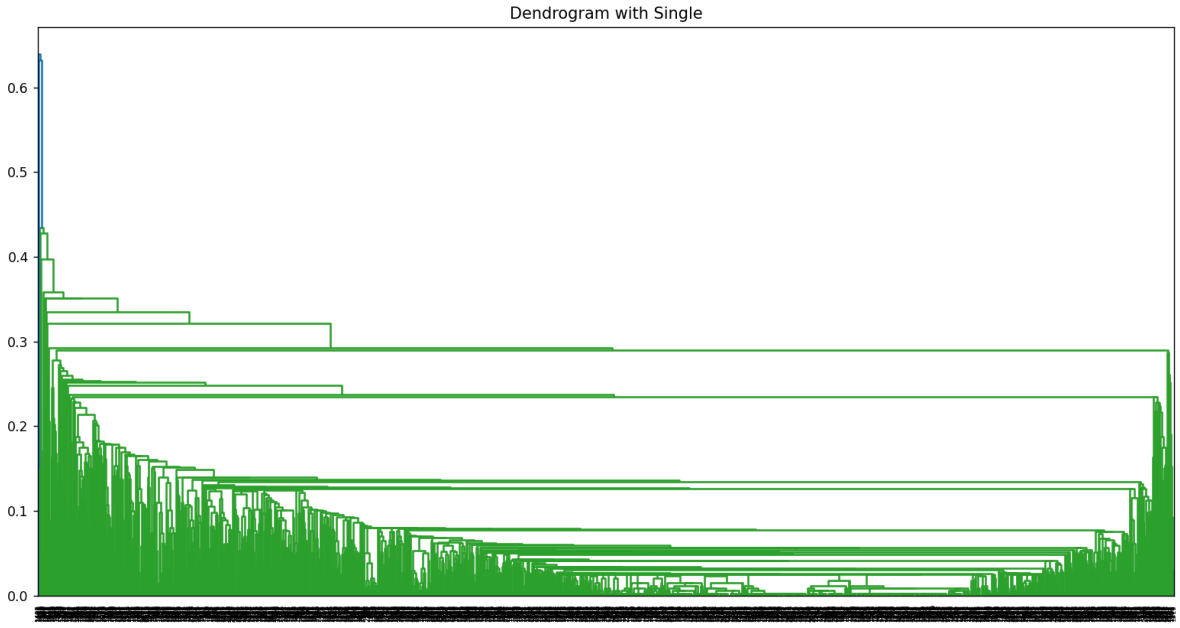


Figure #38 : Single Linkage Dendrogram

We can see the Single Linkage dendrogram in the Figure #38

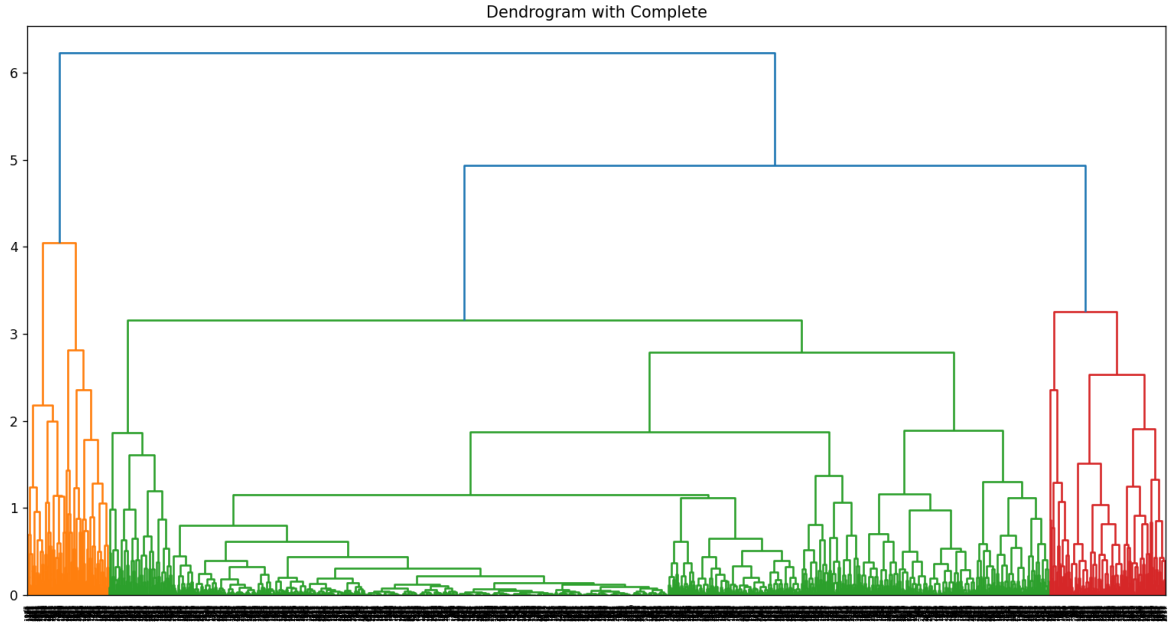


Figure #39 : Complete Linkage Dendrogram

➢ 6.3 Apply one density-based clustering algorithm:

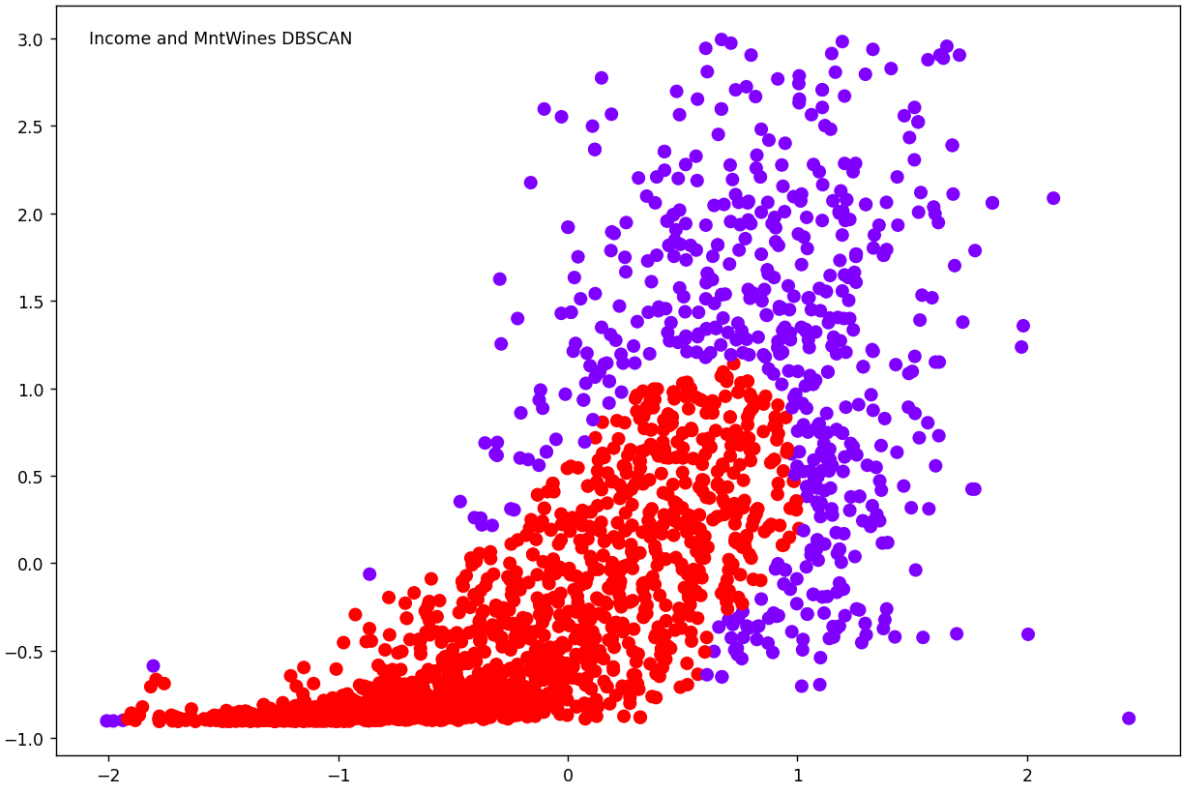


Figure #40 : DBSCAN Graph

As we can see in the **Figure #40** DBSCAN does not work like the K means because it does not group the samples according to a line. It calculates the differences between the samples and uses a special technique.

7. Ensemble Learning

➢ 7.1 Select a classification algorithm to predict a target attribute:

We selected the Decision Tree Classification Algorithm which predicts the "Response" attribute. We selected this classification because of its low accuracy compared to the others so that we can increase it.

➢ 7.2 Apply bagging with the selected algorithm:

When creating a bagging ensemble, we give the Decision Tree as the base estimator and oob score as True so we can calculate training sets accuracy. In **Figure#41** We first calculated the Decision Tree classifier then training the Bagging Ensemble the result increased significantly. By placing the testing samples accuracy went up just by a little margin. This means that the training set did not cause overfitting. By doing this our data is more robust to noisy data. Each created model gives an equal weighted prediction. In **Figure#42** it is also possible to see the decision trees that went into the aggregated classifier individually.

```
Accuracy of classification:
0.8337837837837838
Accuracy of bagging classifier (training set):
0.8806666666666667
Accuracy of bagging classifier (testing set - to see if it overfits):
0.8905405405405405
```

Figure #41 : Accuracy of Bagging Ensemble on DecisionTree

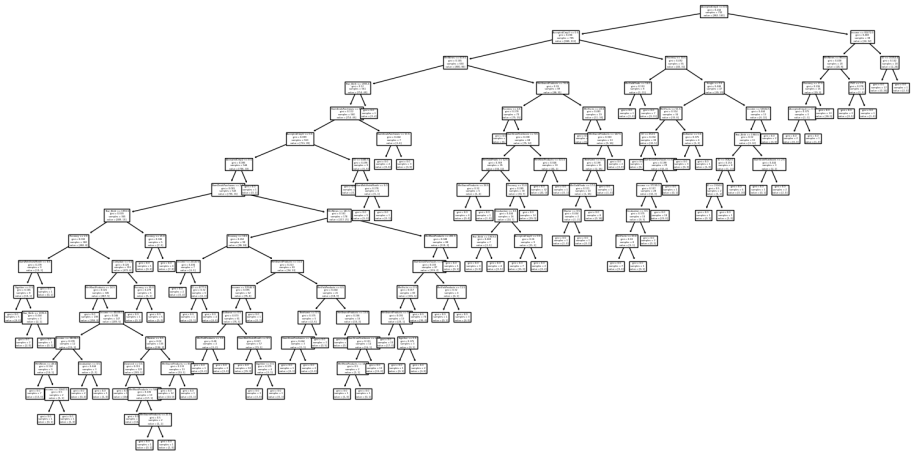


Figure #42

➢ **7.3 Apply AdaBoost with the selected algorithm:**

When applying AdaBoost we again give the Decision Tree as the base estimator. First we calculated the accuracy of both the Decision Tree Classifier's accuracy and then the ensemble method AdaBoost's accuracy. As you can see in **Figure #43** The increase in it is fairly low so we should change the training set to get better results.

In AdaBoost each sample starts with the same weight. As the iteration progresses the failure sample's weight increases and the success' weight decreases so the chance of the failures to get selected can get higher. After the last iteration the votes are collected. In figures number **#44** and **#45** We can observe the Boosting on both the testing and training set.

```
Accuracy of classification:
0.8351351351351352
Accuracy of AdaBoost Ensemble:
0.8432432432432433
```

Figure #43 : Accuracy of AdaBoost Ensemble on DecisionTree

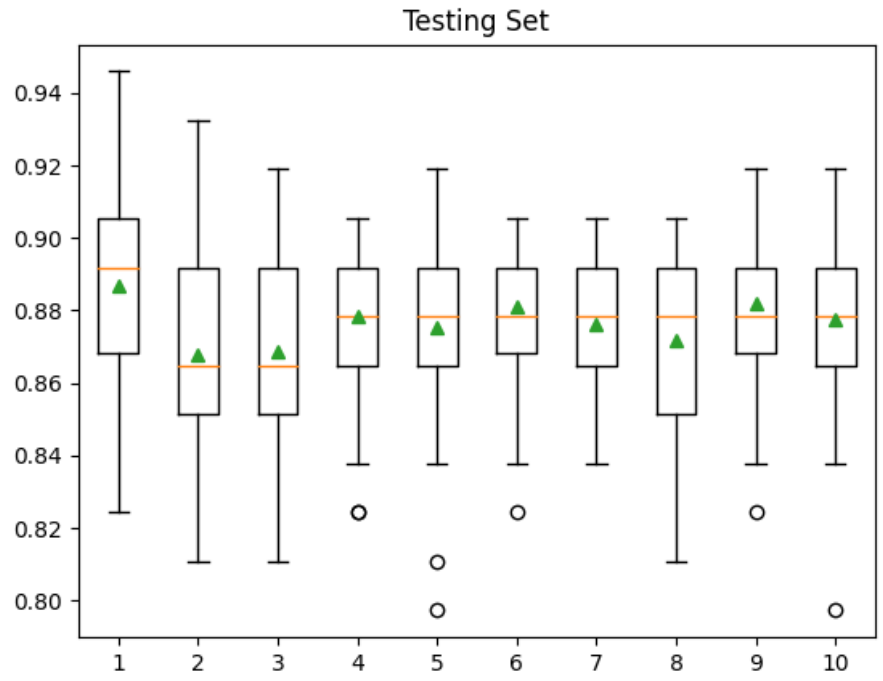


Figure #44: AdaBoost ensemble on Testing Set

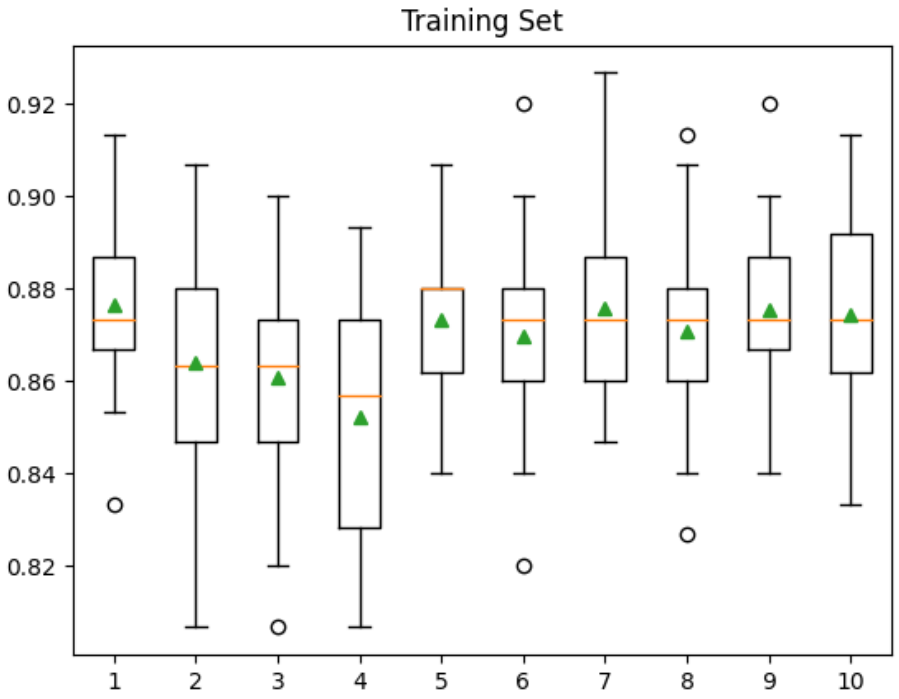


Figure #45: AdaBoost ensemble on Training Set

➢ 7.4 Train a Random Forest:

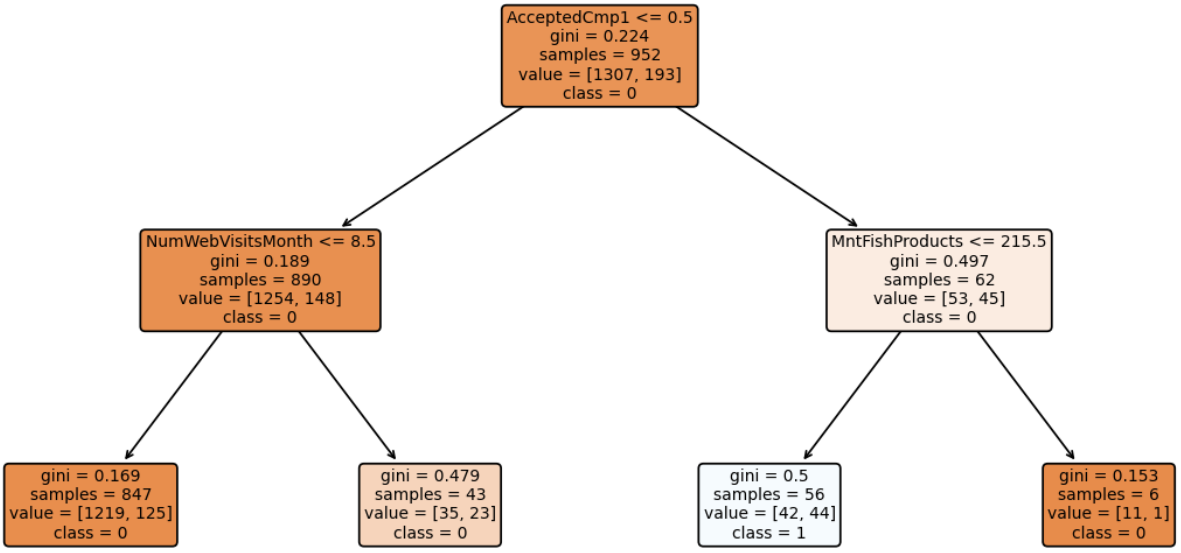


Figure #46

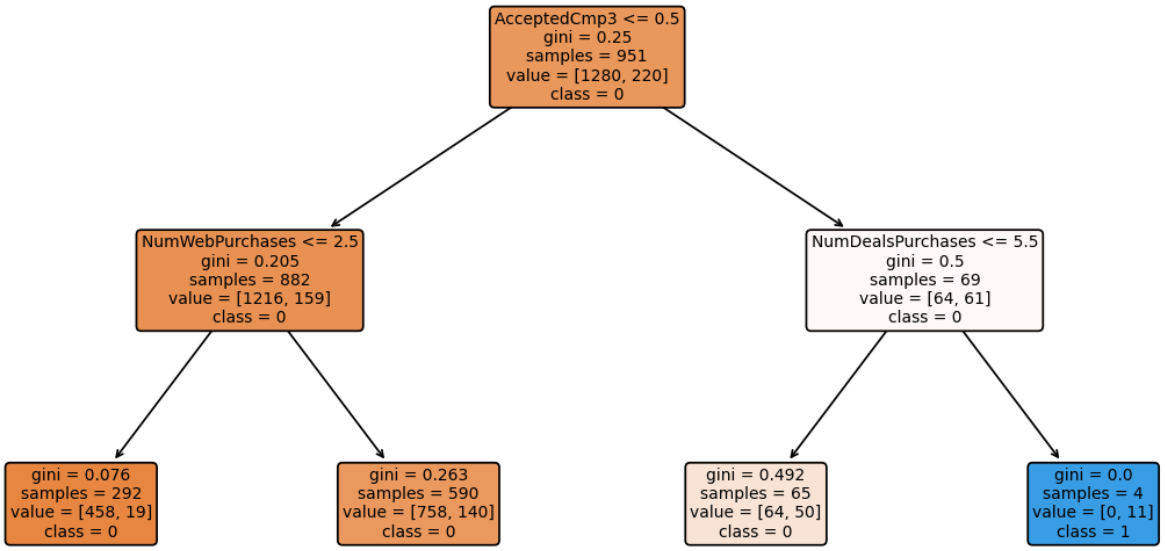


Figure #47

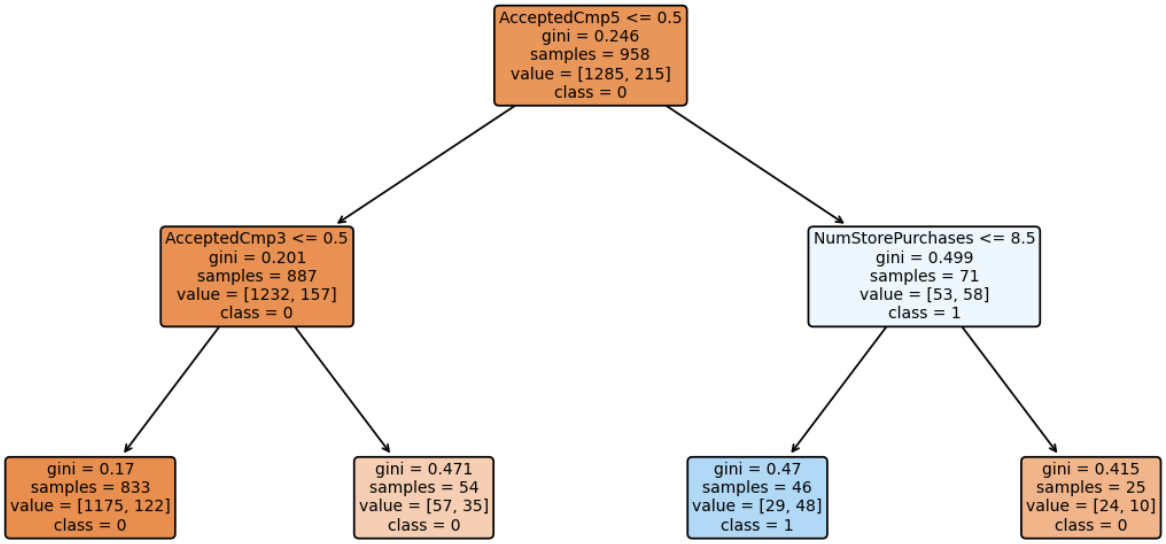


Figure #48

As you can see above. **Figure #46, Figure #47** and **Figure #48** are the 3 decision trees which are shown from a RandomForestClassifier algorithm. With just three estimators, it's clear how scaling up gives a rich, diverse representation of the knowledge that can be successfully assembled into a highly-accurate model.(We used ⅓ of the data as Testing data set and ⅔ for Training data.)

Note: *The more trees there are in the forest, the more varied the model can be. There is a point of diminishing returns, however, because with many trees fit on a random subset of features, there will be a fair amount of similar trees in the ensemble that don't offer much diversity, and which will start to have too much voting power and skew the ensemble to be overfit on the training dataset, hurting generalisation to the validation set.*

```
----------------------------
Accuracy
0.8824324324324324
```

Figure #49 : Accuracy of RandomForestClassifier

We used RandomForestClassifier with the same Train and Test data. Now we have the accuracy 0.88, as you can see from the **Figure #49.**

➢ 7.5 Compare the performance of the methods:

As we can see, when we used different approaches, we got different accuracy percentages. For example when we use Decision Tree we got 0.83, when we used bagging it gave us 0.88, when we used Random Forest Classifier we got 0.88 and lastly the AdaBoost gave us an accuracy which constantly changes (sometimes lower than Decision Tree, other times higher) in here is 0.84. According to the comparison we can see that using bagging technique improved the result in a very significant way which even is proven by showing that the percentage got closer to the Random Forest Classifier. So Instead of using a simple Decision Tree algorithm. Using bagging or a Random Forest Classifier is so much better. The classifier we used in AdaBoost was poor so the new accuracy value was not something to note for, we should abandon it and train a new training set to get better testing set values.

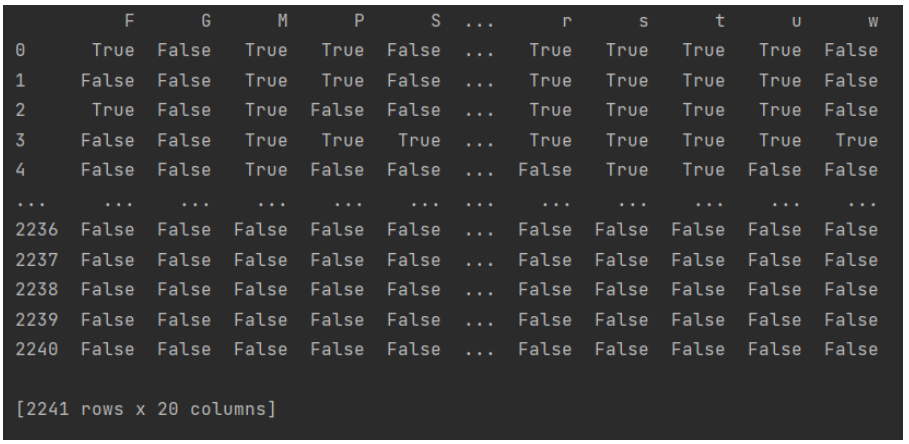
8. Bonus: Association Mining

➢ 8.1 Apply Apriori algorithm to your data set:

Since association mining is one of the first techniques used in data mining, it is one of the first analyses that comes to mind when data mining is mentioned. Association mining is an approach that supports future studies by analysing past data and detecting association behaviours in this data.

The purpose here is to find the association relationship between the products purchased by the customers during shopping and to determine the purchasing habits of the customers in line with this relationship data. In this way, sellers achieve profit maximisation by providing effective and profitable marketing thanks to these discovered association relations and habits.

Apriori Algorithm, which has an iterative (repetitive) nature, is the most common algorithm used for association mining.



	F	G	M	P	S	...	r	s	t	u	w
0	True	False	True	True	False	...	True	True	True	True	False
1	False	False	True	True	False	...	True	True	True	True	False
2	True	False	True	False	False	...	True	True	True	True	False
3	False	False	True	True	True	...	True	True	True	True	True
4	False	False	True	False	False	...	False	True	True	False	False
...	...	...	...	...	...	...	...	...	...	...	...
2236	False	False	False	False	False	...	False	False	False	False	False
2237	False	False	False	False	False	...	False	False	False	False	False
2238	False	False	False	False	False	...	False	False	False	False	False
2239	False	False	False	False	False	...	False	False	False	False	False
2240	False	False	False	False	False	...	False	False	False	False	False

[2241 rows x 20 columns]

Figure #50 : True-False array

We transformed the dataset into a True-False array form as you can see in the Figure #50

9. Conclusion

First of all, we have seen that by using the preprocessing techniques we actually prepare our data to be used in the machine learning algorithms. The methods we used in the first section were useful in sections 2 and 3.

Secondly, splitting data into training and testing sets were useful to train our machine learning algorithm then by using the test set we actually tested if the prediction is successful and its accuracy is high enough to be useful.

Thirdly, although we got the high value of accuracy we have learned that there are some techniques to improve the performance of the machine learning algorithms and we used these techniques in the third section and found out that instead of using a single classification or regression algorithm we can support it in different ways or we can use some other alternatives.

Finally, in our report we have covered each of the topics that we have learned in the lecture and the codes that we have implemented during the semester are in the github repository.