Kerim Sever
Denzel
Mcbrian
Yuktaben Shah
Karen Wu
IS 465001 - Data Mining Project

PART 1. Introduction

The data can be found here:
https://www.kaggle.com/imakash3011/customer-personality-analysis .

The data we are using shows the demographics, items bought, and other activities of customers within a store within 2 years. We are conducting data analysis with this data to find any insightful trends that inform customer buying behavior and what this store can do to increase profit. We are using algorithms to find which target market our advertisement group should focus on. By using different methods we will be able to say confidently which group is better for targeting sales and which groups we should focus less on.

PART 2. Data

The total number of records is 2240. The total number of attributes is 31.
The attributes of this dataset are:

Personal Demographics

- **ID:** Customer's unique identifier
- **Year_Birth:** Customer's birth year
- **Education:** Customer's education level
- **Marital_Status:** Customer's marital status
- **Income:** Customer's yearly household income
- **Kidhome:** Number of children in customer's household
- **Teenhome:** Number of teenagers in customer's household
- **Dt_Customer:** Date of customer's enrollment with the company
- **Recency:** Number of days since customer's last purchase
- **Complain:** 1 if the customer complained in the last 2 years, 0 otherwise

Money Spent on Products

- **MntWines:** Amount spent on wine in last 2 years
- **MntFruits:** Amount spent on fruits in last 2 years

- **MntMeatProducts:** Amount spent on meat in last 2 years
- **MntFishProducts:** Amount spent on fish in last 2 years
- **MntSweetProducts:** Amount spent on sweets in last 2 years
- **MntGoldProds:** Amount spent on gold in last 2 years

Promotional Activity

- **NumDealsPurchases:** Number of purchases made with a discount
- **AcceptedCmp1:** 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- **AcceptedCmp2:** 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- **AcceptedCmp3:** 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- **AcceptedCmp4:** 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- **AcceptedCmp5:** 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- **Response:** 1 if customer accepted the offer in the last campaign, 0 otherwise

Avenues for Buying Behavior

- **NumWebPurchases:** Number of purchases made through the company's website
- **NumCatalogPurchases:** Number of purchases made using a catalogue
- **NumStorePurchases:** Number of purchases made directly in stores
- **NumWebVisitsMonth:** Number of visits to company's website in the last month

Here is a snippet of the dataset in Excel:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	ID	Year_Birt	Education	Marital_St	Income	Kidhome	Teenhom	Dt_Custor	Recency	MntWine	MntFruits	MntMeat	MntFishP	MntSwee	MntGoldP	NumDeal	NumWeb	NumCatal	NumStore	NumWeb	Accepted	Accepted	Accepted	Accepted	Accepted	Complain	Z_CostCor	Z_Revenu	Response
2	5524	1957	Graduatio	Single	58138	0	0	4/9/2012	58	635	88	546	172	88	88	3	8	10	4	7	0	0	0	0	0	0	3	11	1
3	2174	1954	Graduatio	Single	46344	1	1	8/3/2014	38	11	1	6	2	1	6	2	1	1	2	5	0	0	0	0	0	0	3	11	0
4	4141	1965	Graduatio	Together	71613	0	0	21-08-201	26	426	49	127	111	21	42	1	8	2	10	4	0	0	0	0	0	0	3	11	0
5	6182	1984	Graduatio	Together	26646	1	0	########	26	11	4	20	10	3	5	2	2	0	4	6	0	0	0	0	0	0	3	11	0
6	5324	1981	PhD	Married	58293	1	0	19-01-201	94	173	43	118	46	27	15	5	5	3	6	5	0	0	0	0	0	0	3	11	0
7	7446	1967	Master	Together	62513	0	1	9/9/2013	16	520	42	98	0	42	14	2	6	4	10	6	0	0	0	0	0	0	3	11	0
8	965	1971	Graduatio	Divorced	55635	0	1	13-11-201	34	235	65	164	50	49	27	4	7	3	7	6	0	0	0	0	0	0	3	11	0
9	6177	1985	PhD	Married	33454	1	0	8/5/2013	32	76	10	56	3	1	23	2	4	0	4	8	0	0	0	0	0	0	3	11	0
10	4855	1974	PhD	Together	30351	1	0	6/6/2013	19	14	0	24	3	3	2	1	3	0	2	9	0	0	0	0	0	0	3	11	1
11	5899	1950	PhD	Together	5648	1	1	13-03-201	68	28	0	6	1	1	13	1	1	0	0	20	1	0	0	0	0	0	3	11	0
12	1994	1983	Graduatio	Married		1	0	15-11-201	11	5	5	6	0	2	1	1	1	0	2	7	0	0	0	0	0	0	3	11	0
13	387	1976	Basic	Married	7500	0	0	13-11-201	59	6	16	11	11	1	16	1	2	0	3	8	0	0	0	0	0	0	3	11	0
14	2125	1959	Graduatio	Divorced	63033	0	0	15-11-201	82	194	61	480	225	112	30	1	3	4	8	2	0	0	0	0	0	0	3	11	0
15	8180	1952	Master	Divorced	59354	1	1	15-11-201	53	233	2	53	3	5	14	3	6	1	5	6	0	0	0	0	0	0	3	11	0
16	2569	1987	Graduatio	Married	17323	0	0	########	38	3	14	17	6	1	5	1	1	0	3	8	0	0	0	0	0	0	3	11	0
17	2114	1946	PhD	Single	82800	0	0	24-11-201	23	1006	22	115	59	68	45	1	7	6	12	3	0	0	1	1	0	0	3	11	1
18	9736	1980	Graduatio	Married	41850	1	1	24-12-201	51	53	5	19	2	13	4	3	3	0	3	8	0	0	0	0	0	0	3	11	0
19	4939	1946	Graduatio	Together	37760	0	0	31-08-201	20	84	5	38	150	12	28	2	4	1	6	7	0	0	0	0	0	0	3	11	0
20	6565	1949	Master	Married	76995	0	1	28-03-201	91	1012	80	498	0	16	176	2	11	4	9	5	0	0	0	1	0	0	3	11	0
21	2278	1985	2n Cycle	Single	33812	1	0	########	86	4	17	19	30	24	39	2	2	1	3	6	0	0	0	0	0	0	3	11	0
22	9360	1982	Graduatio	Married	37040	0	0	8/8/2012	41	86	2	73	69	38	48	1	4	2	5	8	0	0	0	0	0	0	3	11	0
23	5376	1979	Graduatio	Married	2447	1	0	6/1/2013	42	1	1	1725	1	1	1	15	0	28	0	1	0	0	0	0	0	0	3	11	0
24	1993	1949	PhD	Married	58607	0	1	23-12-201	63	867	0	86	0	0	19	3	2	3	9	8	0	1	0	0	0	0	3	11	0
25	4047	1954	PhD	Married	65324	0	1	########	0	384	0	102	21	32	5	3	6	2	9	4	0	0	0	0	0	0	3	11	0
26	1409	1951	Graduatio	Together	40689	0	1	18-03-201	69	270	3	27	39	6	99	7	7	1	5	8	0	0	0	0	0	0	3	11	0
27	7892	1969	Graduatio	Single	18589	0	0	2/1/2013	89	6	4	25	15	12	13	2	2	1	3	7	0	0	0	0	0	0	3	11	0
28	2404	1976	Graduatio	Married	53359	1	1	27-05-201	4	173	4	30	3	6	41	4	5	1	4	7	0	0	0	0	0	0	3	11	0
29	5255	1986	Graduatio	Single		1	0	20-02-201	19	5	1	3	3	263	362	0	27	0	0	1	0	0	0	0	0	0	3	11	0
30	9422	1989	Graduatio	Married	38360	1	0	31-05-201	26	36	2	42	20	21	10	2	2	1	4	3	0	0	0	0	0	0	3	11	0
31	1966	1965	PhD	Married	84618	0	0	22-11-201	96	684	100	801	21	66	0	1	6	9	10	2	0	0	1	0	0	0	3	11	0
32	6864	1989	Master	Divorced	10979	0	0	22-05-201	34	8	4	10	2	2	4	2	3	0	3	5	0	0	0	0	0	0	3	11	0
33	3033	1963	Master	Together	38620	0	0	########	56	112	17	44	34	22	89	1	2	5	3	3	0	0	0	0	0	0	3	11	0
34	5710	1970	Graduatio	Together	40548	0	1	########	31	110	0	5	2	0	3	2	2	1	4	5	0	1	0	0	0	0	3	11	0
35	7373	1952	PhD	Divorced	46610	0	2	29-10-201	8	96	12	96	33	22	43	6	4	1	6	6	0	0	0	0	0	0	3	11	1
36	8755	1946	Master	Married	68657	0	0	20-02-201	4	482	34	471	119	68	22	1	3	5	9	7	0	0	0	0	0	0	3	11	0

PART 3. Method
Clustering Using Apriori by Mcbrian

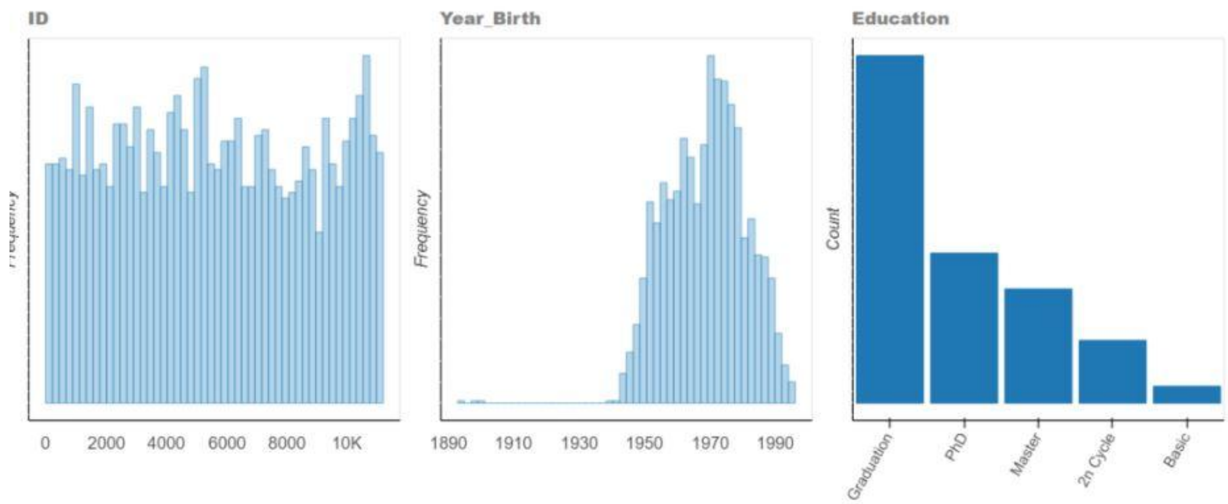
In [9]:
```
import numpy as np
import pandas as pd
import datetime
from datetime import date
from dataprep.eda import plot, plot_correlation, create_report, plot_missing

import matplotlib
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from sklearn.preprocessing import StandardScaler, normalize
from sklearn import metrics
from sklearn.mixture import GaussianMixture
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import warnings
warnings.filterwarnings('ignore')
data=pd.read_excel('final_Excel_Data.xls',header=0,sep=';')
```

```
pd.set_option('display.max_colwidth', 999)
pd.options.display.float_format = "{:.3f}".format
association=data.copy()
df = pd.get_dummies(association)
min_support = 0.08
max_len = 10
frequent_items = apriori(df, use_colnames=True, min_support=min_support, max_len=max_len + 1)
rules = association_rules(frequent_items, metric='lift', min_threshold=1)

product='Wines'
segment='Biggest consumer'
target = '{\'%s_segment_%s\'}' %(product,segment)
results_personnal_care = rules[rules['consequents'].astype(str).str.contains(target, na=False)].sort_values(by='confidence', as
results_personnal_care.head()
```

```
cut_labels = ['Low consumer', 'Frequent consumer', 'Biggest consumer']
data['Wines_segment'] = pd.qcut(data['Wines'][data['Wines']>0],q=[0, .25, .75, 1], labels=cut_labels).astype
data['Fruits_segment'] = pd.qcut(data['Fruits'][data['Fruits']>0],q=[0, .25, .75, 1], labels=cut_labels).ast
data['Meat_segment'] = pd.qcut(data['Meat'][data['Meat']>0],q=[0, .25, .75, 1], labels=cut_labels).astype("o
data['Fish_segment'] = pd.qcut(data['Fish'][data['Fish']>0],q=[0, .25, .75, 1], labels=cut_labels).astype("o
data['Sweets_segment'] = pd.qcut(data['Sweets'][data['Sweets']>0],q=[0, .25, .75, 1], labels=cut_labels).ast
data['Gold_segment'] = pd.qcut(data['Gold'][data['Gold']>0],q=[0, .25, .75, 1], labels=cut_labels).astype("o
data.replace(np.nan, "Non consumer",inplace=True)
data.drop(columns=['Spending','Wines','Fruits','Meat','Fish','Sweets','Gold'],inplace=True)
data = data.astype(object)
```

```python
scaler=StandardScaler()
dataset_temp=data[['Income','Seniority','Spending']]
X_std=scaler.fit_transform(dataset_temp)
X = normalize(X_std,norm='l2')

gmm=GaussianMixture(n_components=4, covariance_type='spherical
labels = gmm.predict(X)
dataset_temp['Cluster'] = labels
dataset_temp=dataset_temp.replace({0:'Stars',1:'Need attentior
data = data.merge(dataset_temp.Cluster, left_index=True, right

pd.options.display.float_format = "{:.0f}".format
summary=data[['Income','Spending','Seniority','Cluster']]
summary.set_index("Cluster", inplace = True)
summary=summary.groupby('Cluster').describe().transpose()
summary.head()
```

```python
data['Spending']=data['MntWines']+data['MntFruits']+data['MntMeatProducts']+data['MntFishProducts']+data['MntSweetProducts']+dat
#Seniority variable creation
last_date = date(2014,10, 4)
data['Seniority']=pd.to_datetime(data['Dt_Customer'], dayfirst=True,format = '%Y-%m-%d')
data['Seniority'] = pd.to_numeric(data['Seniority'].dt.date.apply(lambda x: (last_date - x)).dt.days, downcast='integer')/30
data=data.rename(columns={'NumWebPurchases': "Web",'NumCatalogPurchases':'Catalog','NumStorePurchases':'Store'})
data['Marital_Status']=data['Marital_Status'].replace({'Divorced':'Alone','Single':'Alone','Married':'In couple','Together':'In
data['Education']=data['Education'].replace({'Basic':'Undergraduate','2n Cycle':'Undergraduate','Graduation':'Postgraduate','Ma:

data['Children']=data['Kidhome']+data['Teenhome']
data['Has_child'] = np.where(data.Children> 0, 'Has child', 'No child')
data['Children'].replace({3: "3 children",2:'2 children',1:'1 child',0:"No child"},inplace=True)
data=data.rename(columns={'MntWines': "Wines",'MntFruits':'Fruits','MntMeatProducts':'Meat','MntFishProducts':'Fish','MntSweetP


data=data[['Age','Education','Marital_Status','Income','Spending','Seniority','Has_child','Children','Wines','Fruits','Meat','F
data.head()
```




```
pd.set_option('display.max_colwidth', 999)
pd.options.display.float_format = "{:.3f}".format
association=data.copy()
df = pd.get_dummies(association)
min_support = 0.08
max_len = 10
frequent_items = apriori(df, use_colnames=True, min_support=min_support, max_len=max_len + 1)
rules = association_rules(frequent_items, metric='lift', min_threshold=1)

product='Wines'
segment='Biggest consumer'
target = '{\'%s_segment_%s\'}' %(product,segment)
results_personnal_care = rules[rules['consequents'].astype(str).str.contains(target, na=False)].sort_values(by='confidence', a
results_personnal_care.head()
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	con
28190	(Cluster_x_Stars, Income_group_High income, Cluster_Stars)	(Wines_segment_Biggest consumer)	0.121	0.249	0.084	0.697	2.800	0.054	
7970	(Cluster_x_Stars, Income_group_High income)	(Wines_segment_Biggest consumer)	0.121	0.249	0.084	0.697	2.800	0.054	
8500	(Cluster_y_Stars, Income_group_High income)	(Wines_segment_Biggest consumer)	0.121	0.249	0.084	0.697	2.800	0.054	
49205	(Cluster_x_Stars, Cluster_y_Stars, Income_group_High income, Cluster_Stars)	(Wines_segment_Biggest consumer)	0.121	0.249	0.084	0.697	2.800	0.054	
8632	(Income_group_High income, Cluster_Stars)	(Wines_segment_Biggest consumer)	0.121	0.249	0.084	0.697	2.800	0.054	

Based on these results, we can conclude that customers who buy the most wine have an average household income of about $70,000 typically buy a lot of meat and have been with the company for 21 months or have some kind of graduate degree.

Principal Component Analysis (PCA) by Karen Wu

In [36]:
```
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
```

In [37]:
```
import pandas as pd
import numpy as np
```

In [38]:
```
df=pd.read_csv('final_Excel_Data2.csv')
```

In [39]:
```
df.head()
```

Out[39]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	Acce
0	5524	1957	Graduation	Single	58138.0	0	0	4/9/2012	58	635	...	7	0	
1	2174	1954	Graduation	Single	46344.0	1	1	8/3/2014	38	11	...	5	0	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	4	0	
3	6182	1984	Graduation	Together	26646.0	1	0	10/2/2014	26	11	...	6	0	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	5	0	

5 rows × 29 columns

In [40]:
```
df.columns
```

Out[40]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response'],
      dtype='object')
```

In [41]:
```
df1=df.drop(['Dt_Customer','ID'], axis=1).reset_index(drop=True)
```

In [42]:
```
df2=pd.get_dummies(df1)
```

df2.head()

In [43]: df2.head()

Above, I am cleaning the data by getting rid of the date they became a customer, their unique ID. This was necessary because these types of data are not a value, and it's very hard to apply general statistics to these. Also, I felt that these types of data do not contribute much when applying principal component analysis.

For other categorical data such as education and marital status, I turned each type of data into its own column, and then each observation would be assigned 0 (they don't have that type of category) or 1 (they have that type of category). So for example, for education, there is a column called Education_Master representing master's degrees. If a person has a master's degree, they would be assigned a 1 for that column, and if they do not have that, they would be assigned a 0.

In [43]:
```
df2.head()
```

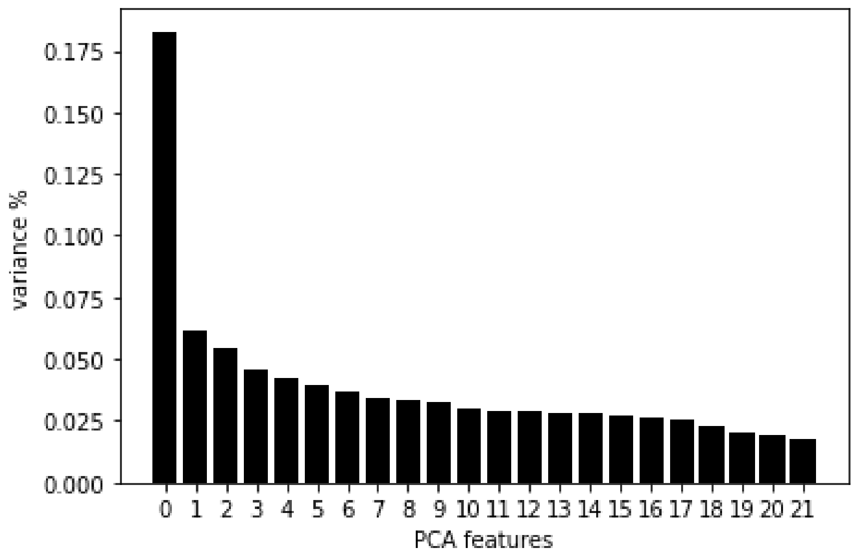
Out[43]:

	Year_Birth	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	...	Education_Master	Educ
0	1957	58138.0	0	0	58	635	88	546	172	88	...	0	
1	1954	46344.0	1	1	38	11	1	6	2	1	...	0	
2	1965	71613.0	0	0	26	426	49	127	111	21	...	0	
3	1984	26646.0	1	0	26	11	4	20	10	3	...	0	
4	1981	58293.0	1	0	94	173	43	118	46	27	...	0	

5 rows × 38 columns

In [44]:
```
from sklearn.decomposition import PCA
df3=df2.dropna()
```

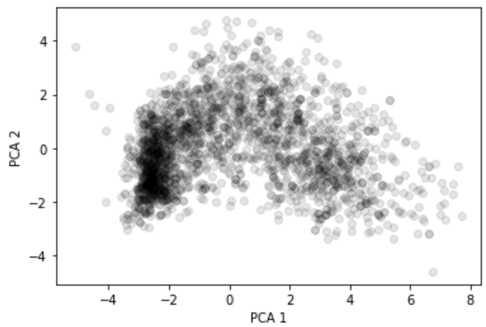
In [45]:
```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot as plt

X_std=StandardScaler().fit_transform(df3)
#Create PCA instance: pca
pca=PCA(n_components=.85)
principalComponents=pca.fit_transform(X_std)

#Plot the variances
features=range(pca.n_components_)
plt.bar(features, pca.explained_variance_ratio_, color="black")
plt.xlabel("PCA features")
plt.ylabel("variance %")
plt.xticks(features)

#Save components to Dataframe
PCA_components=pd.DataFrame(principalComponents)
```



Now, I am creating a bar chart with Principal Component Analysis. It shows which combination of attributes in the Excel sheet contribute significantly to 85% of the variance of the entire dataset. According to the above bar chart, Combination 0 explains over 17.5% of the variance within 85% of the variance of the entire dataset. And so on.

In [46]:
```
plt.scatter(PCA_components[0], PCA_components[1], alpha=.1, color="black")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
```

Out[46]: `Text(0, 0.5, 'PCA 2')`



Here is the relationship between the first two Principal Components (the two components that contribute most to the variance within Combination 0, PCA 1 and PCA 2). Considering the black group of dots located towards the left, most observations of these two Principal components tend to be directly and negatively correlated.

In [98]:
```
np.sort(-np.abs(pca.components_[0,:]))
```

Out[98]:
```
array([-0.31755624, -0.31298904, -0.30504322, -0.29158166, -0.28205084,
       -0.27493274, -0.26784037, -0.26499622, -0.25437016, -0.24624593,
       -0.22097937, -0.21193774, -0.19130708, -0.17072746, -0.10792021,
       -0.09587867, -0.06613795, -0.05925583, -0.05849151, -0.05433463,
       -0.05428814, -0.02755342, -0.02552083, -0.02301645, -0.01938084,
       -0.01524039, -0.01300057, -0.01123162, -0.00969995, -0.00953974,
       -0.00908339, -0.00457948, -0.00215665, -0.00145377, -0.000991  ,
       -0.00046776, -0.        , -0.        ])
```

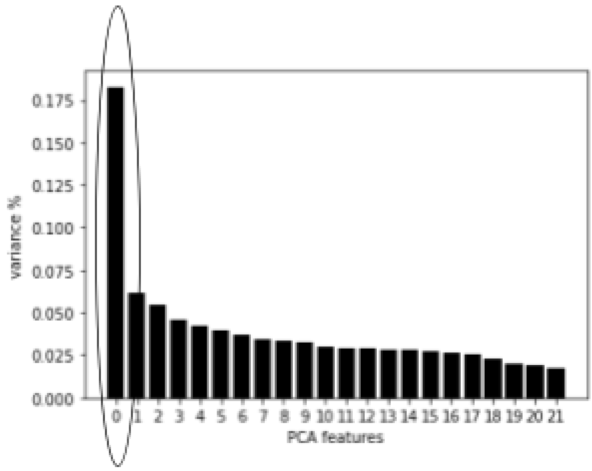
In [103]:
```
most_imp=np.argsort(-np.abs(pca.components_[0,:]))
```

In [104]:
```
df3.columns[most_imp]
```

Out[104]:
```
Index(['NumCatalogPurchases', 'MntMeatProducts', 'MntWines', 'Income',
       'NumStorePurchases', 'MntFishProducts', 'MntSweetProducts', 'MntFruits',
       'Kidhome', 'NumWebVisitsMonth', 'MntGoldProds', 'NumWebPurchases',
       'AcceptedCmp5', 'AcceptedCmp1', 'Response', 'AcceptedCmp4',
       'Education_Basic', 'Year_Birth', 'AcceptedCmp2', 'NumDealsPurchases',
       'Teenhome', 'Education_Graduation', 'Marital_Status_Widow',
       'Marital_Status_Absurd', 'AcceptedCmp3', 'Complain',
       'Education_2n Cycle', 'Marital_Status_Alone', 'Education_Master',
       'Marital_Status_Married', 'Education_PhD', 'Marital_Status_YOLO',
       'Marital_Status_Divorced', 'Marital_Status_Single', 'Recency',
       'Marital_Status_Together', 'Z_Revenue', 'Z_CostContact'],
      dtype='object')
```

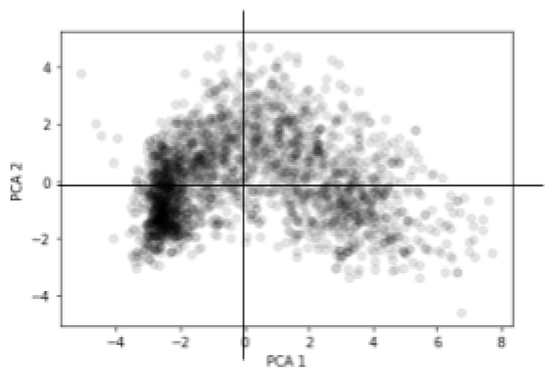
Here, we find a list of categories ordered by the most significant category within Combination 0 to least significant, assuming that the first two entities are PCA 1 and 2. According to the list above, PCA 1 is Numb Catalog Purchases (number of purchases made using a catalogue) and PCA 2 is MntMeatProducts (amount spent on meat in the last 2 years).

1.



Combination 0 (of attributes) contributes the most (over 17.5%) of 85% of the variance within the entire dataset.

2.



The highest contributors of variance within Combination 0 are PCA 1 and PCA 2. They are mainly directly correlated as they decrease with each other.

In summary, **the number of purchases made using a catalog** (NumCatalogPurchases-PCA1) **and the amount spent on meat in the last two years** (MntMeatProducts-PCA 2) are the two categories that **contribute the most to the combination of attributes** (Combination 0) **that contributes the most** (over 17.5%) **to 85% of the variance within the entire dataset.** Considering PCA 1 and PCA 2's general significance to overall variance and that they are directly and decreasingly correlated, one can assume that their quantities tend to drastically vary but people tend to spend less money on meat if they purchase from the store catalog less. One theory that explains this is that people will mainly buy meat when it is on sale/featured in the store catalog.

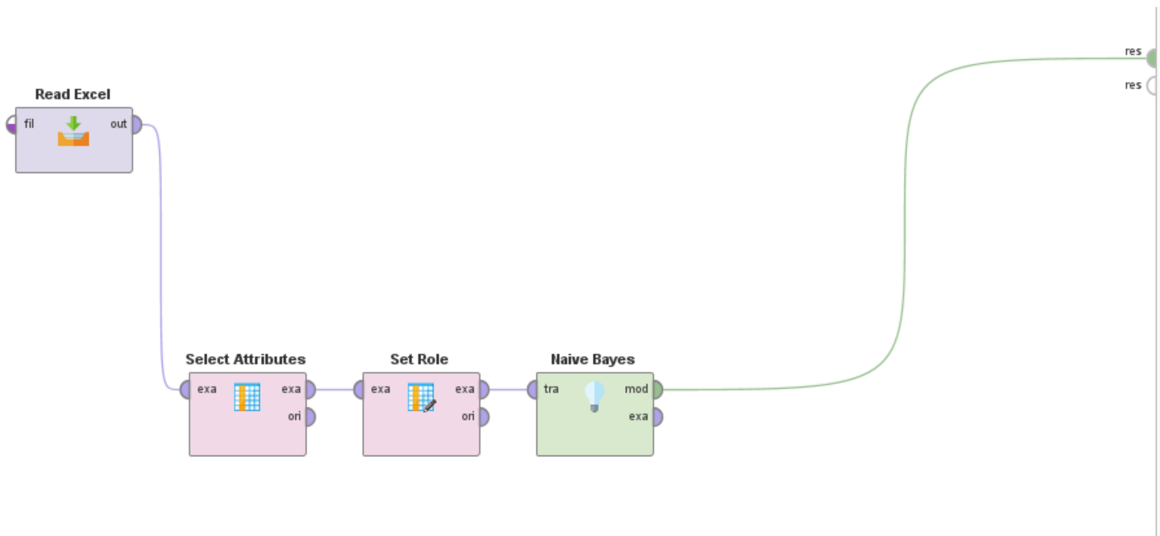
3.

```
In [104]: df3.columns[most_imp]

Out[104]: Index(['NumCatalogPurchases', 'MntMeatProducts', 'MntWines', 'Income',
       'NumStorePurchases', 'MntFishProducts', 'MntSweetProducts', 'MntFruits',
       'Kidhome', 'NumWebVisitsMonth', 'MntGoldProds', 'NumWebPurchases',
       'AcceptedCmp5', 'AcceptedCmp1', 'Response', 'AcceptedCmp4',
       'Education_Basic', 'Year_Birth', 'AcceptedCmp2', 'NumDealsPurchases',
       'Teenhome', 'Education_Graduation', 'Marital_Status_Widow',
       'Marital_Status_Absurd', 'AcceptedCmp3', 'Complain',
       'Education_2n Cycle', 'Marital_Status_Alone', 'Education_Master',
       'Marital_Status_Married', 'Education_PhD', 'Marital_Status_YOLO',
       'Marital_Status_Divorced', 'Marital_Status_Single', 'Recency',
       'Marital_Status_Together', 'Z_Revenue', 'Z_CostContact'],
      dtype='object')
```

PCA 1 is NumCatalogPurchases (number of purchases made with a catalog) and PCA 2 is MntMeatProducts (amount spent on meat products within 2 years).

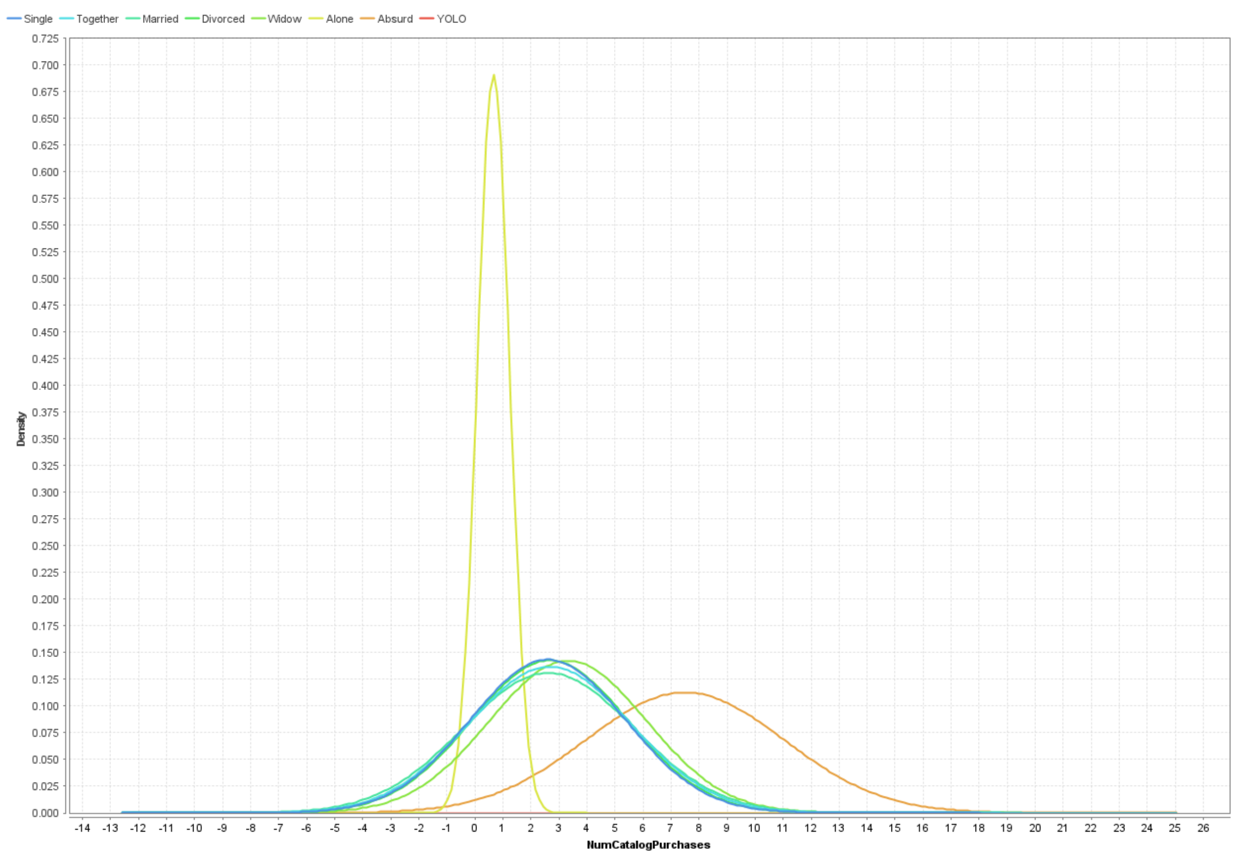
Naive Bayes by Kerim Sever

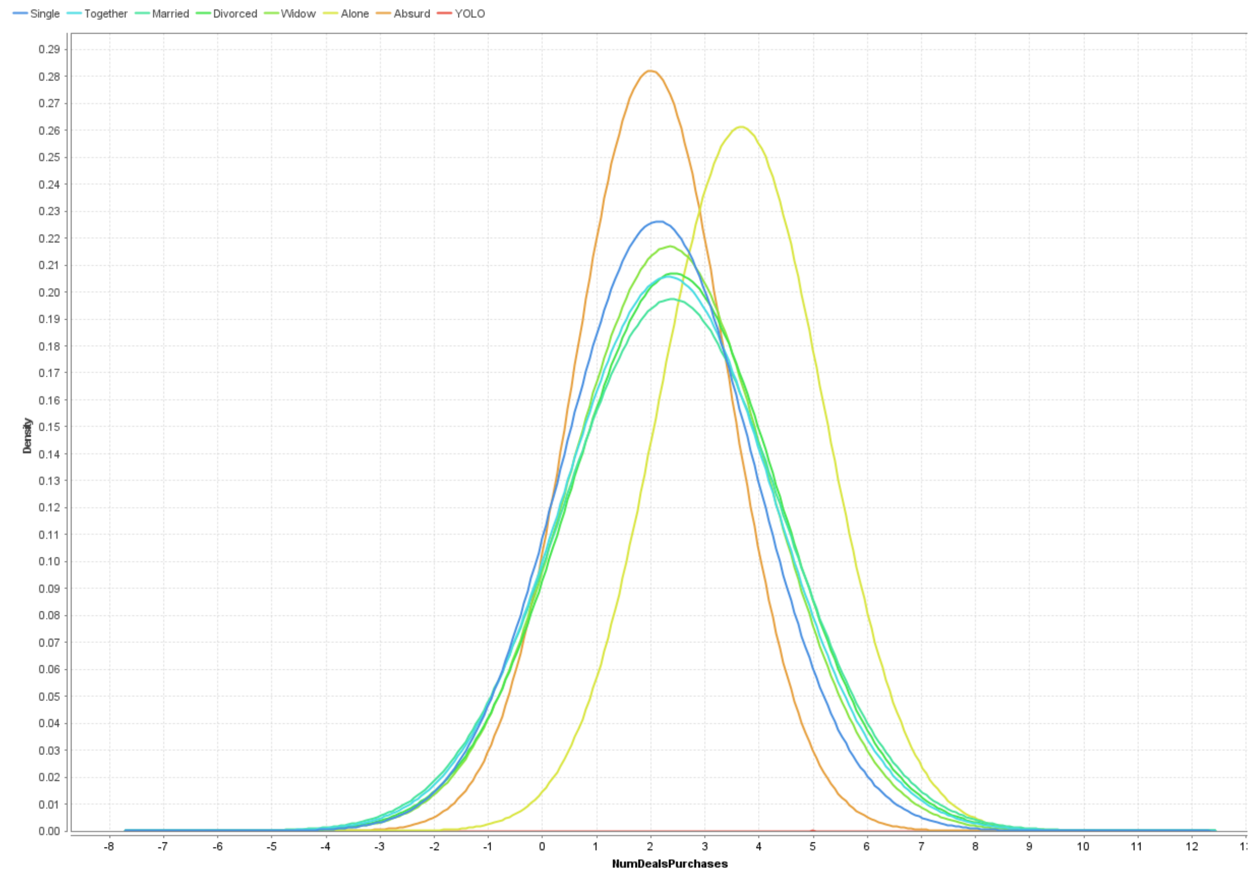
Below I added the CSV file into rapid miner. I excluded ID and the other labels besides Income, Kidhome, Parameter, single, married, yolo, divorced, widow, alone, and absurd. Once I imported the data I needed to select my attributes, I filtered subsets and added marital status, kid home income, NumCatalogPurchases, NumDealsPurchases, NumStoresPurchases, and NumWebPurchases. Then when I set my role I made my attribute status equal to Marital status as my label for the set role. Then I linked that Naive Bayes with Laplace correction and ran my data.

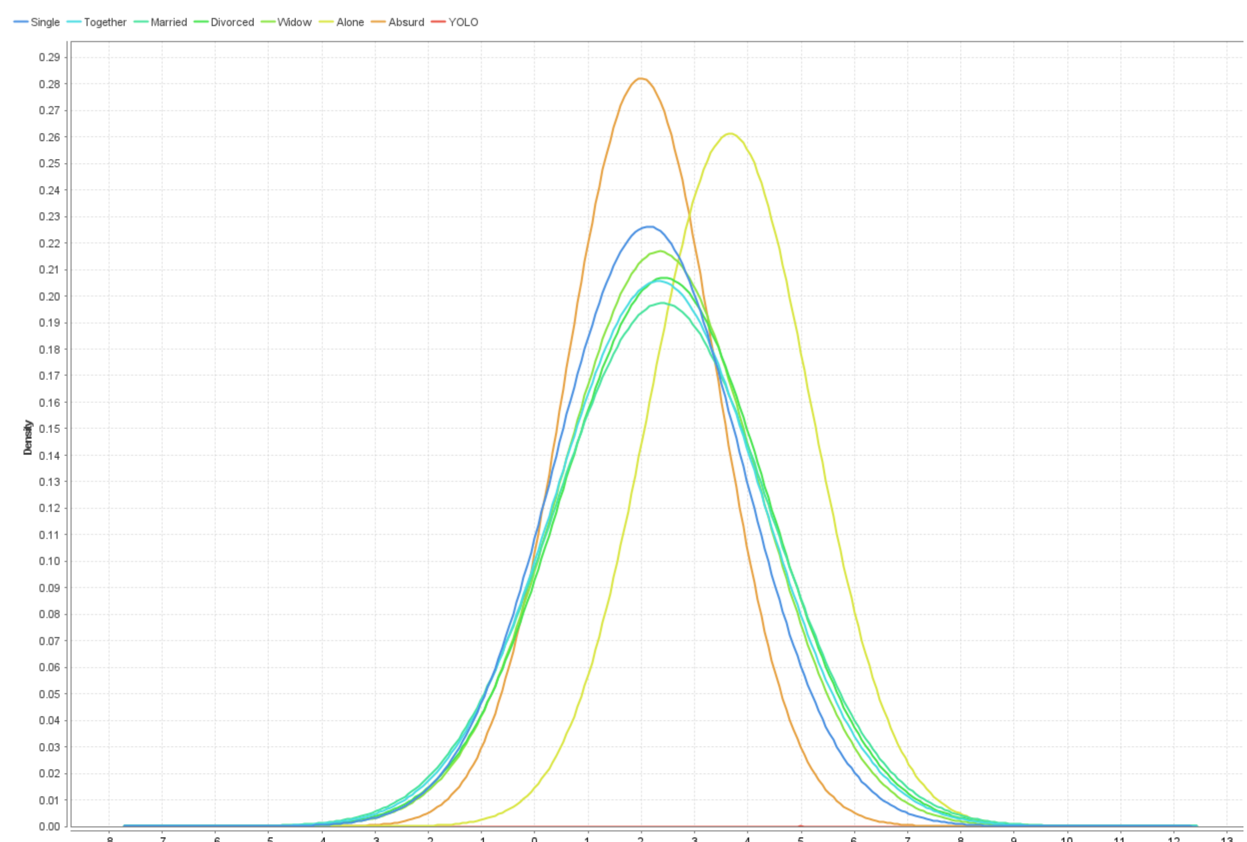


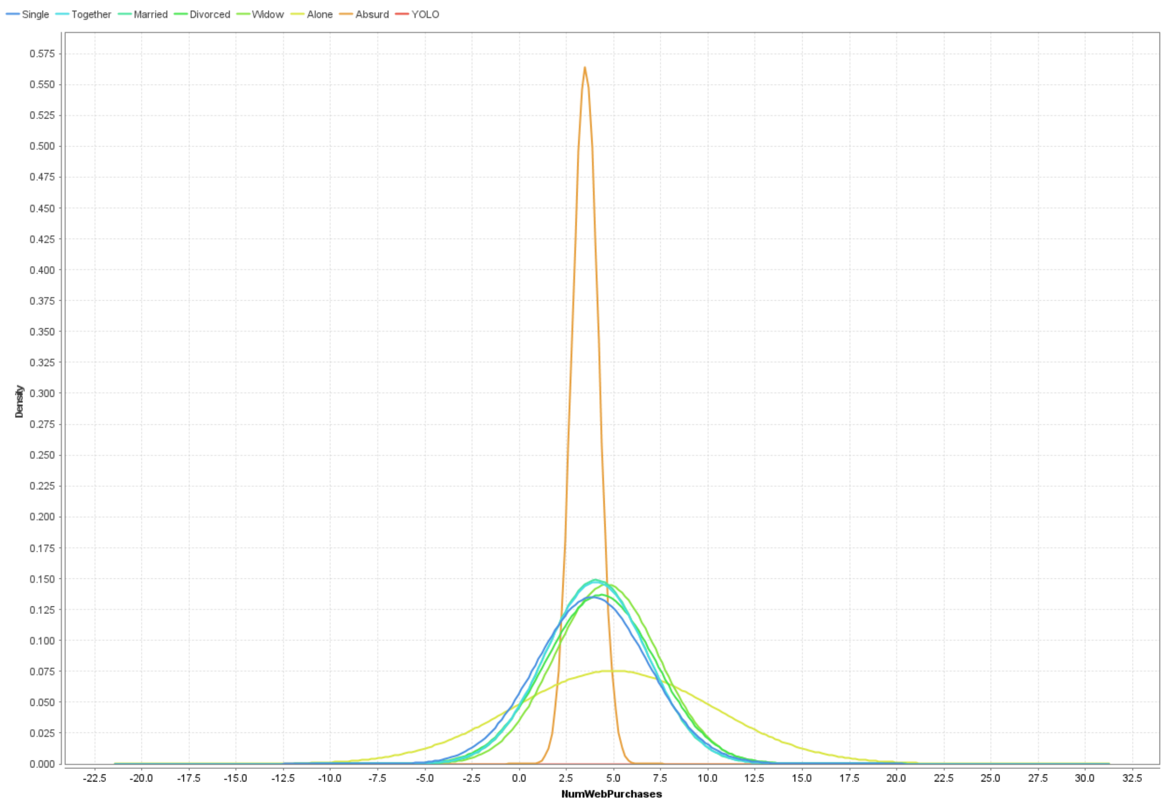
This is the results that I got through naive bayes, as we can see the incomes of people that have a small means of a kid home usually have a higher income then married or together. I was surprised to see alone with a kid mean of 1 but the “alone” data shows as a horizontal line when I filter the graph to kid home. In the charts below we can see that the yolo has a higher average from 50k-75k with most of the survey sample being within that range. We can tell that households with less kids have more spending money and households with children are spread out across the graph. Marketing companies should target households with little to no children because they will most likely have more money to spend on themselves. They can also target these households with more expensive items because they will have a surplus of money compared to family households. Overall households with less people/no kids, typically spend more money online, at stores, deal purchases, catalog and marketers should advertise more towards them to sell products.

Attribute	Parameter	Single	Together	Married	Divorced	Widow	Alone	Absurd	YOLO
Income	mean	50995.350	53245.534	51724.979	52834.228	56481.553	43789	72365.500	48432
Income	standard deviation	22229.542	33644.101	21449.406	21239.760	16837.952	15215.133	9727.668	0.001
Kidhome	mean	0.465	0.450	0.456	0.414	0.234	1	0	0
Kidhome	standard deviation	0.543	0.538	0.545	0.527	0.426	0.001	0.001	0.001
NumDealsPurchases	mean	2.131	2.324	2.392	2.435	2.338	3.667	2	5
NumDealsPurchases	standard deviation	1.763	1.940	2.021	1.928	1.840	1.528	1.414	0.001
NumWebPurchases	mean	3.873	4.081	4.088	4.310	4.623	5	3.500	7
NumWebPurchases	standard deviation	2.952	2.710	2.678	2.916	2.748	5.292	0.707	0.001
NumCatalogPurchases	mean	2.600	2.676	2.625	2.672	3.325	0.667	7.500	1
NumCatalogPurchases	standard deviation	2.780	2.914	3.045	2.794	2.802	0.577	3.536	0.001
NumStorePurchases	mean	5.640	5.736	5.851	5.819	6.416	4	6.500	6
NumStorePurchases	standard deviation	3.255	3.221	3.255	3.327	3.278	2	0.707	0.001

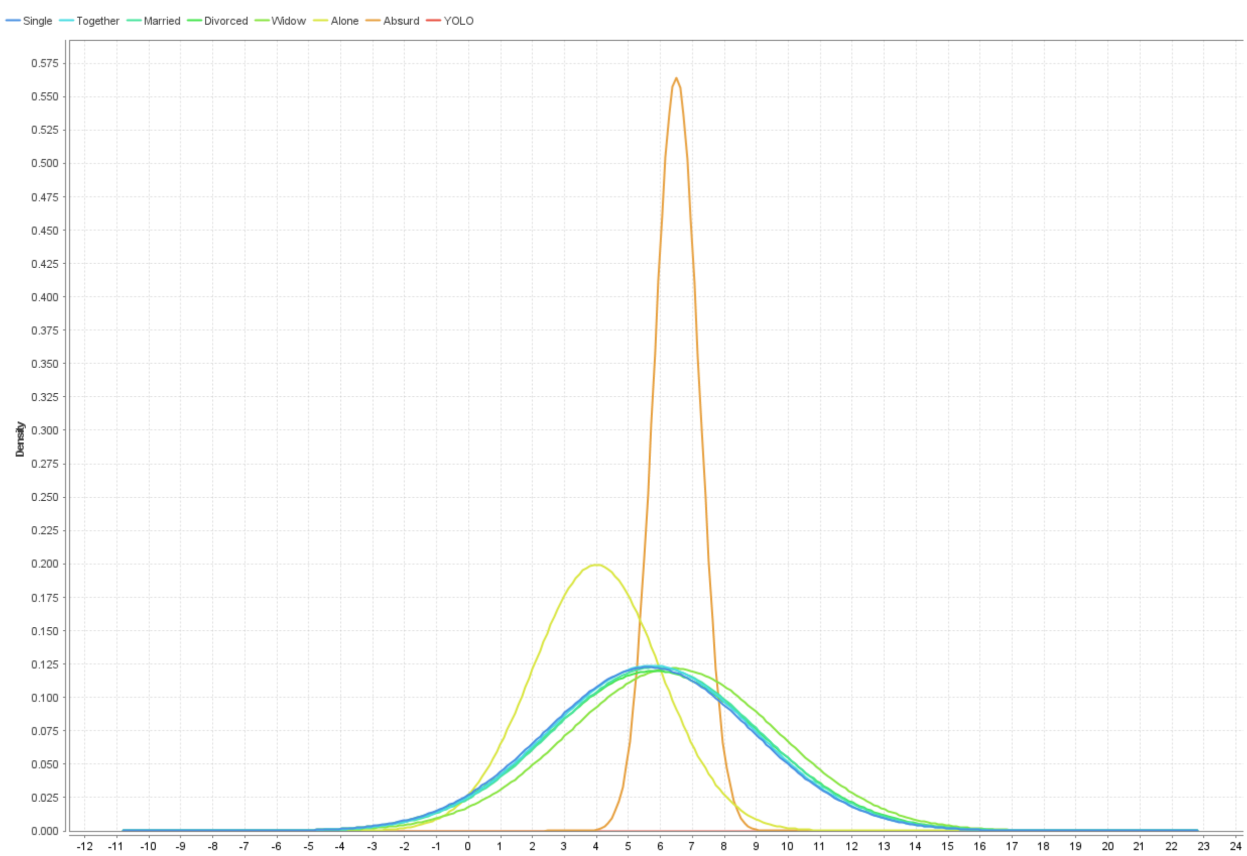



Single Together Married Divorced Widow Alone Absurd YOLO
Density
NumDealsPurchases


Single Together Married Divorced Widow Alone Absurd YOLO
Density

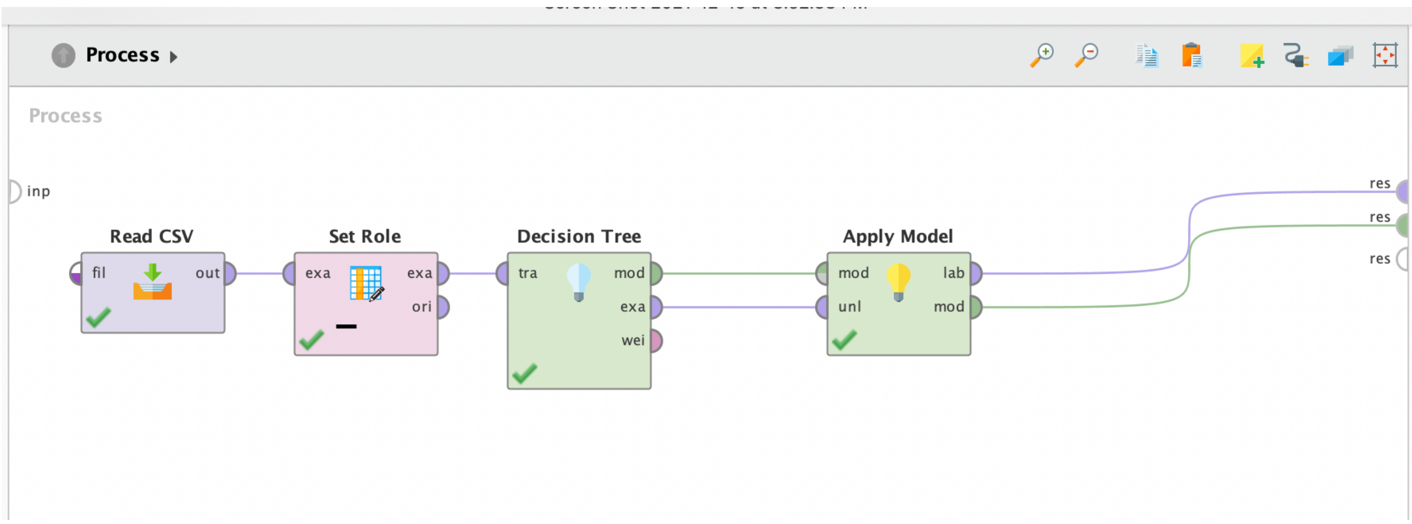


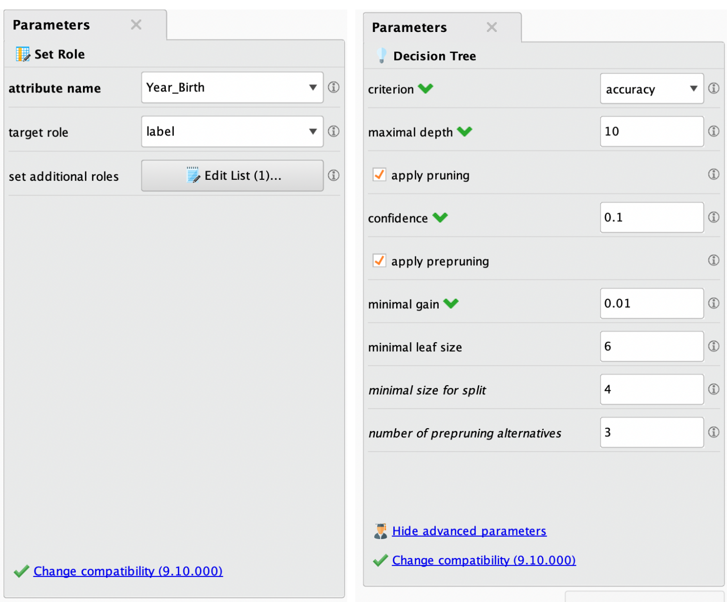
Single
Together
Married
Divorced
Widow
Alone
Absurd
YOLO
Density
NumWebPurchases



Single
Together
Married
Divorced
Widow
Alone
Absurd
YOLO
Density
NumStorePurchases

Decision Trees by Yuktaben Shah



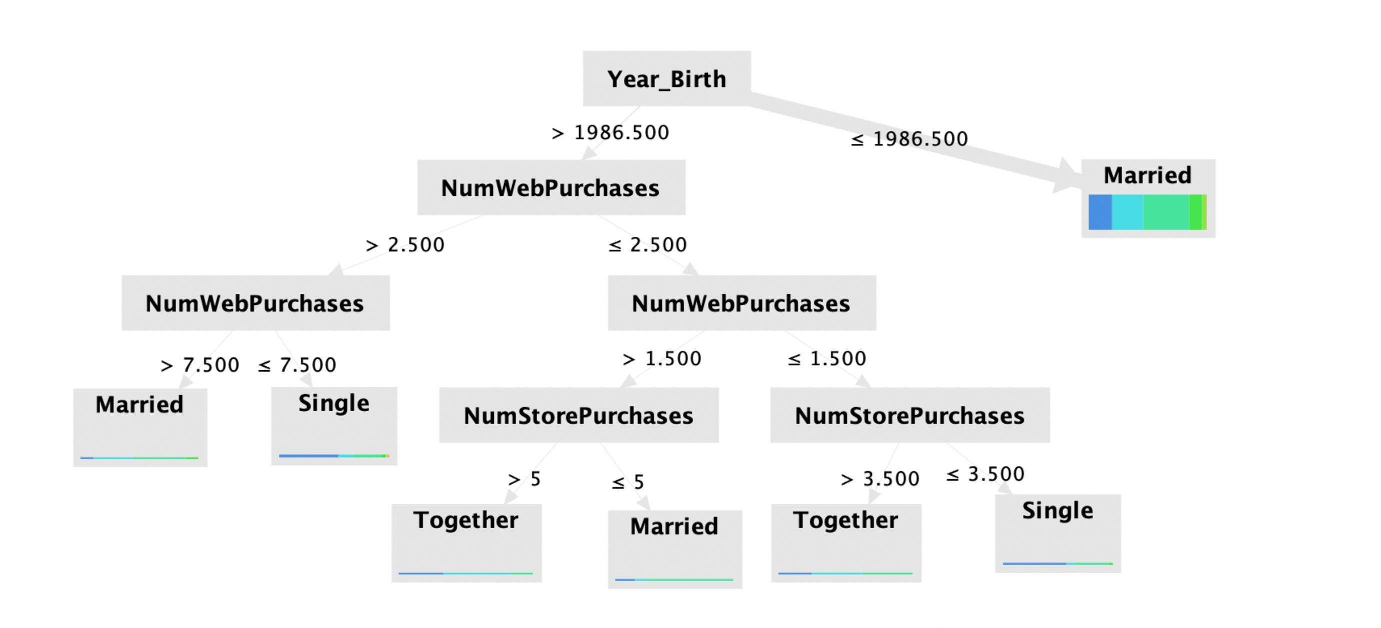


Above, using the Read CSV operator, I started with importing the CSV data file. The attributes I included are marital status (single, together, married, divorced, widow, alone, absurd, and YOLO), num web purchases, Num store Purchases, and year birth. I excluded the other attributes since I didn't need them. Then, after importing all the data, I used the set role parameters to set the year birth as attribute name and target role as a label. Next, I connected that to the decision tree operator and set its parameters — the criterion as accuracy and minimal leaf size as 6. Then, I joined the decision tree operator to the apply mode operator. At last, I made the final connection and ran the process to get the output.

Tree

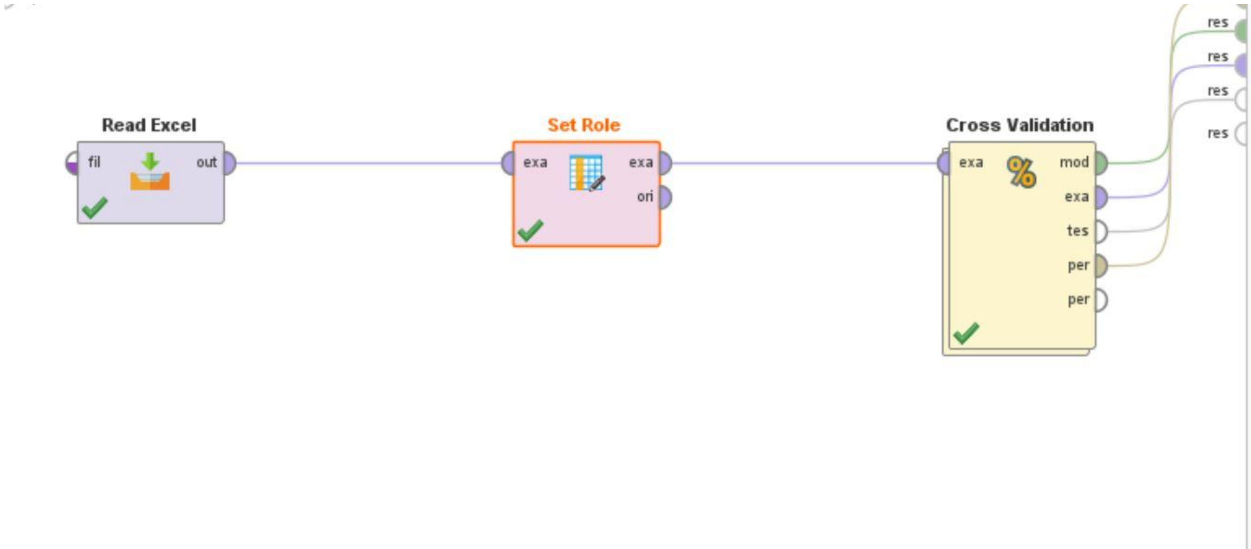
```
Year_Birth > 1986.500
|   NumWebPurchases > 2.500
|   |   NumWebPurchases > 7.500: Married {Single=1, Together=3, Married=4, Divorced=1, Widow=0, Alone=0, Absurd=0, YOLO=0}
|   |   NumWebPurchases ≤ 7.500: Single {Single=43, Together=11, Married=20, Divorced=4, Widow=0, Alone=1, Absurd=1, YOLO=0}
|   NumWebPurchases ≤ 2.500
|   |   NumWebPurchases > 1.500
|   |   |   NumStorePurchases > 5: Together {Single=2, Together=3, Married=1, Divorced=0, Widow=0, Alone=0, Absurd=0, YOLO=0}
|   |   |   NumStorePurchases ≤ 5: Married {Single=3, Together=2, Married=13, Divorced=0, Widow=0, Alone=0, Absurd=0, YOLO=0}
|   |   NumWebPurchases ≤ 1.500
|   |   |   NumStorePurchases > 3.500: Together {Single=2, Together=3, Married=3, Divorced=0, Widow=0, Alone=0, Absurd=0, YOLO=0}
|   |   |   NumStorePurchases ≤ 3.500: Single {Single=15, Together=2, Married=8, Divorced=1, Widow=0, Alone=0, Absurd=0, YOLO=0}
Year_Birth ≤ 1986.500: Married {Single=414, Together=556, Married=815, Divorced=226, Widow=77, Alone=2, Absurd=1, YOLO=2}
```

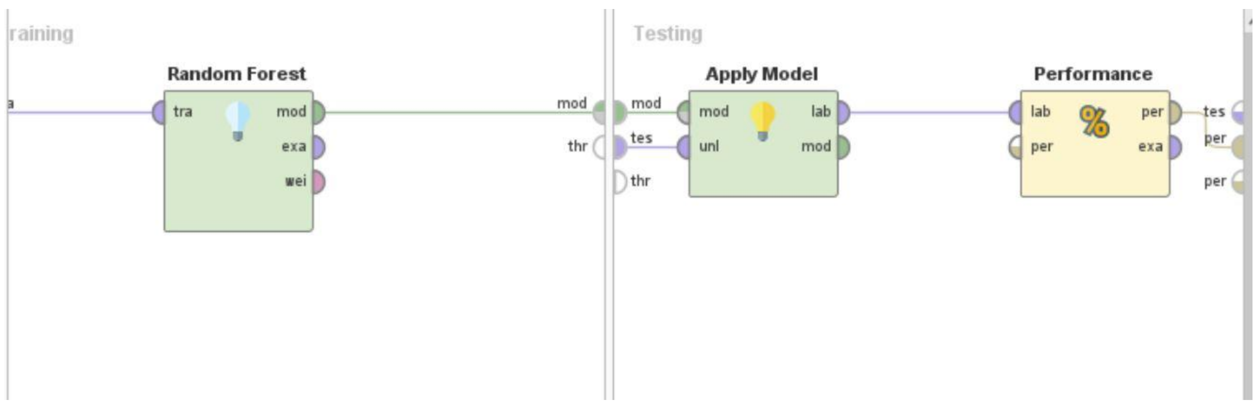
Based on the tree's description above, you can see that single people make significantly more web purchases than different marital statuses. To my surprise, single people also make more store purchases compared to married people and people who are together. However, singles are much more likely to make web purchases than store purchases.



Above, the decision tree is a visualization of the data and description. It compares how people with different marital statuses are likely to purchase through different places such as store or web purchases.

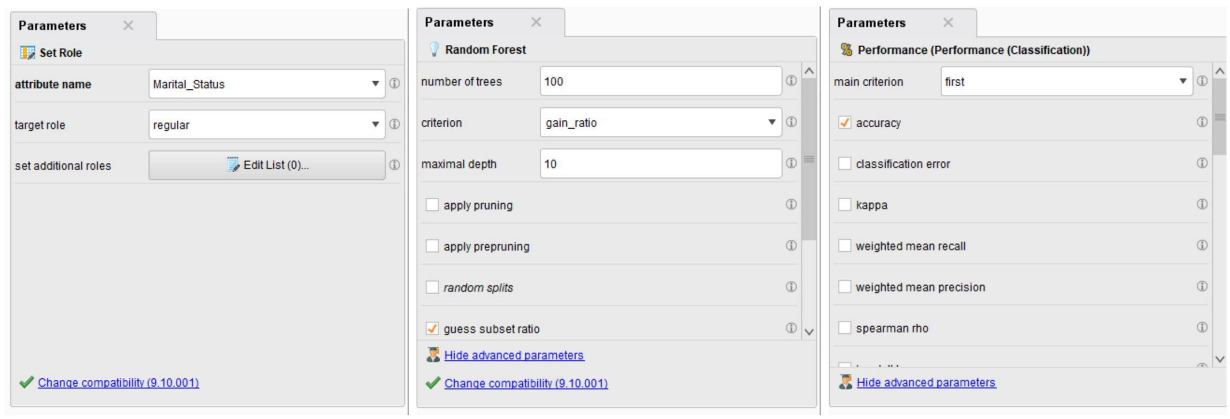
Random Forest by Denzel Tovar
Work Flow





I decided to use rapid miner to show my results for using the random forest algorithm. In the above images you can see the work-flow and also observe that I used a “read excel” operator to read my data.

Parameters



Once my data was ready to be read I connected it to a “set role” operator where the attribute I chose was marital status. Once my attributes were set I went to set up my random forest operators and set it’s parameters to a total of 100 trees with a maximal depth of 10.Furthermore, I connected it to an apply model operator and then a “performance(classification)” operator where its parameter was set to accuracy. Once I had all my operators set I ran my rapid miner and I had no errors.

```
NumWebVisitsMonth > 19.500: PhD {Graduation=0, PhD=6, Master=0, Basic=0, 2n Cycle=0}
NumWebVisitsMonth ≤ 19.500
|   MntWines > 0.500
|   |   Income = ?
|   |   |   MntMeatProducts > 983: 2n Cycle {Graduation=0, PhD=0, Master=0, Basic=0, 2n Cycle=2}
|   |   |   MntMeatProducts ≤ 983
|   |   |   |   MntFruits > 0.500
|   |   |   |   |   MntMeatProducts > 277.500: Master {Graduation=0, PhD=0, Master=1, Basic=0, 2n Cycle=0}
|   |   |   |   |   MntMeatProducts ≤ 277.500
|   |   |   |   |   |   NumWebVisitsMonth > 8.500: PhD {Graduation=0, PhD=1, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   NumWebVisitsMonth ≤ 8.500
|   |   |   |   |   |   |   Kidhome > 0.500
|   |   |   |   |   |   |   |   NumWebVisitsMonth > 7.500: 2n Cycle {Graduation=1, PhD=0, Master=0, Basic=0, 2n Cycle=2}
|   |   |   |   |   |   |   |   NumWebVisitsMonth ≤ 7.500: Graduation {Graduation=8, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   |   Kidhome ≤ 0.500
|   |   |   |   |   |   |   |   MntWines > 209: Graduation {Graduation=2, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   |   |   MntWines ≤ 209: Master {Graduation=0, PhD=0, Master=1, Basic=0, 2n Cycle=0}
|   |   |   |   MntFruits ≤ 0.500
|   |   |   |   |   MntGoldProds > 85.500: Graduation {Graduation=1, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   MntGoldProds ≤ 85.500
|   |   |   |   |   |   MntGoldProds > 19.500: Master {Graduation=0, PhD=0, Master=1, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   MntGoldProds ≤ 19.500: PhD {Graduation=0, PhD=4, Master=0, Basic=0, 2n Cycle=0}
|   |   Income > 28284.500
|   |   |   NumWebVisitsMonth > 9.500: Master {Graduation=0, PhD=0, Master=1, Basic=0, 2n Cycle=0}
|   |   |   NumWebVisitsMonth ≤ 9.500
|   |   |   |   NumDealsPurchases > 12.500: Master {Graduation=0, PhD=0, Master=3, Basic=0, 2n Cycle=0}
|   |   |   |   NumDealsPurchases ≤ 12.500
|   |   |   |   |   Income > 160065: PhD {Graduation=0, PhD=2, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   Income ≤ 160065
|   |   |   |   |   |   MntSweetProducts > 194.500: Graduation {Graduation=6, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   MntSweetProducts ≤ 194.500
|   |   |   |   |   |   |   NumWebVisitsMonth > 0.500
|   |   |   |   |   |   |   |   NumStorePurchases > 1.500: Graduation {Graduation=968, PhD=424, Master=308, Basic=1, 2n Cycle=151}
|   |   |   |   |   |   |   |   NumStorePurchases ≤ 1.500: Master {Graduation=0, PhD=0, Master=2, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   |   NumWebVisitsMonth ≤ 0.500
|   |   |   |   |   |   |   |   MntFruits > 9: PhD {Graduation=0, PhD=3, Master=0, Basic=0, 2n Cycle=1}
|   |   |   |   |   |   |   |   MntFruits ≤ 9: Graduation {Graduation=1, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   Income ≤ 28284.500
|   |   |   NumWebVisitsMonth > 0.500
|   |   |   |   MntFishProducts > 37.500: Graduation {Graduation=11, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   MntFishProducts ≤ 37.500
|   |   |   |   |   NumStorePurchases > 5.500: Master {Graduation=0, PhD=0, Master=2, Basic=0, 2n Cycle=0}
|   |   |   |   |   NumStorePurchases ≤ 5.500
|   |   |   |   |   |   MntGoldProds > 173.500
|   |   |   |   |   |   |   Marital_Status = Married: Graduation {Graduation=2, PhD=0, Master=0, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   |   Marital_Status = Single: Master {Graduation=0, PhD=0, Master=3, Basic=0, 2n Cycle=0}
|   |   |   |   |   |   MntGoldProds ≤ 173.500
```

The above image displays one of the tree models that was created from my random forest algorithm. In the above model we are observing the total number of web visits in a single month. As you can see in the above tree model, individuals with higher income are more than likely to visit the website more times. Those with an income higher than $28,204 averaged out to visit the store website more than 9.5 times a month. In comparison to those who make less than average out to visit the website less than one time in total.

accuracy: 50.89% +/- 0.82% (micro average: 50.89%)

	true Graduation	true PhD	true Master	true Basic	true 2n Cycle	class precision
pred. Graduation	1115	481	368	38	191	50.84%
pred. PhD	4	4	1	0	1	40.00%
pred. Master	1	0	0	0	2	0.00%
pred. Basic	6	0	0	16	4	61.54%
pred. 2n Cycle	1	1	1	0	5	62.50%
class recall	98.94%	0.82%	0.00%	29.63%	2.46%	

In regards to my confusion matrix it had an accuracy of 50.89 percent where it was able to predict true graduation the most efficiently. In conclusion, those with higher income are more likely to make purchases in almost all of the categories. In which it can be seen the most in wine and gold purchased which are more luxury items which are not frequently bought by those in lower income.

PART 4. Conclusion

PART IV. Conclusion
(to be completed as a team):

-**Which evaluation metrics should be used (Recall or precision) and why**
The evaluation metric that we should be using is precision. Precision evaluation is equal to true positive over actual results, which is the percentage of total results that is classified by our algorithms. In order to give an accurate answer to the marketing firm that we are coming with, analysis for precision would be best to help them focus on the single target market. We would show that target markets that are single and have less kids are more likely to spend more.

-**Which algorithm works best, at what parameter setup**
We felt Naive best was a good algorithm because it is a faster algorithm for multi-class attributes. Since we were trying to discover customer behavior given the various attributes we had, it was the right fit.

-What does the finding tell you?

From our findings we can conclude that most households that have little to no kid home average and do not classify under married or together are more likely to make more purchases and advertising companies should target that market.

Can you take action to improve the performance and solve a problem based on the findings?

Yes, actions can be taken to improve the performance and solve the problem.

Do you have any recommended actions?

Yes. We would suggest marketing tailors its advertisements, discounts and reward programs to incentivise more people in the target group to buy. We discovered since unmarried people earning an average income of $65k, then they should be the focus of any marketing campaign.