report

February 11, 2019

#
Analyzing customers for targeted marketing

0.1 1.) Introduction

0.2 2.) Dataset

0.3 3.) Exploratory Data Analysis

0.4 4.) Data Wrangling

0.5 5.) Machine Learning

0.6 6.) Conclusion

###
Introduction

An e-commerce clothing brand that wishes to break into a new market. Data is collected from a sample of potential customers from that market. Data includes behavioral spending patterns. The goal of this project is to cluster customers so the e-commerce company can target those customers who are more likely to spend on clothes. Lastly, we also have unseen data to test our algorithm. ###
Dataset

```
In [13]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns

         df = pd.read_csv("millenial_market_research.csv")

In [2]: df.head()

Out[2]:     Age  Gender  Music  Movies/Theaters  Tech/Gadgets  Museums  Food/Dining  \
        0  17.0    male    7.3              8.1           2.8      1.6          4.5
        1  21.0  female    9.4              9.3           2.2      2.2          3.2
        2  19.0  female    6.8              7.5           6.4      2.1          7.8
```

```
3  26.0  female    4.5               6.8            1.3      8.5          8.0
4  19.0  female    9.1               9.8            1.4      3.9          3.1

   Camping/Hiking  Concerts  Clubs/Dancing       ...         Art  Shopping  \
0             7.1       0.3            0.6       ...         2.9       9.9
1             9.5       5.4            1.3       ...         0.6       3.0
2             4.4       1.9            5.8       ...         4.7       4.4
3             7.4       1.5            6.2       ...         1.1       1.8
4             5.4       8.2            4.7       ...         4.5       4.4

   Social Media  Reading  Socializing  Gaming  Entertainment Spending  \
0           4.4      1.4          8.7     6.7                     9.7
1           5.7      4.5          5.3     9.5                     5.3
2           2.7      8.5         10.0     3.5                     3.5
3           5.5      9.8          5.1     7.0                     3.5
4           5.1      5.0          6.8     3.0                     2.0

   Clothing Spending  Internet Spending  Retail Spending
0                8.6                9.5              8.2
1                3.5                4.2             10.0
2                5.2                0.1              3.1
3                2.3                0.3              7.2
4                5.9                5.6              8.2

[5 rows x 23 columns]
```

###
Exploratory Data Analysis

In [3]:
```
df.count()
```

Out[3]:
```
Age                          10000
Gender                       10000
Music                        10000
Movies/Theaters              10000
Tech/Gadgets                 10000
Museums                      10000
Food/Dining                  10000
Camping/Hiking               10000
Concerts                     10000
Clubs/Dancing                10000
Writing                      10000
Sports                       10000
Gardening                    10000
Art                          10000
Shopping                     10000
Social Media                 10000
Reading                      10000
```

```
Socializing               10000
Gaming                    10000
Entertainment Spending    10000
Clothing Spending         10000
Internet Spending         10000
Retail Spending           10000
dtype: int64
```

In [4]:
```
df.isnull().values.any()
```

Out[4]:
```
False
```

In [5]:
```
df.isna().values.any()
```

Out[5]:
```
False
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 23 columns):
Age                       10000 non-null float64
Gender                    10000 non-null object
Music                     10000 non-null float64
Movies/Theaters           10000 non-null float64
Tech/Gadgets              10000 non-null float64
Museums                   10000 non-null float64
Food/Dining               10000 non-null float64
Camping/Hiking            10000 non-null float64
Concerts                  10000 non-null float64
Clubs/Dancing             10000 non-null float64
Writing                   10000 non-null float64
Sports                    10000 non-null float64
Gardening                 10000 non-null float64
Art                       10000 non-null float64
Shopping                  10000 non-null float64
Social Media              10000 non-null float64
Reading                   10000 non-null float64
Socializing               10000 non-null float64
Gaming                    10000 non-null float64
Entertainment Spending    10000 non-null float64
Clothing Spending         10000 non-null float64
Internet Spending         10000 non-null float64
Retail Spending           10000 non-null float64
dtypes: float64(22), object(1)
memory usage: 1.8+ MB
```

In [7]:
```
df.describe()
```

Out[7]:

	Age	Music	Movies/Theaters	Tech/Gadgets	\
count	10000.000000	10000.000000	10000.000000	10000.000000	
mean	20.386600	8.485650	8.221250	4.334410	
std	2.780596	1.401132	1.488302	2.922271	
min	15.000000	0.000000	0.100000	0.000000	
25%	19.000000	8.200000	7.600000	1.700000	
50%	20.000000	8.800000	8.600000	4.000000	
75%	21.000000	9.400000	9.300000	6.700000	
max	30.000000	10.000000	10.000000	10.000000	

	Museums	Food/Dining	Camping/Hiking	Concerts	Clubs/Dancing	\
count	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000	
mean	4.226790	3.51436	6.339660	3.859940	3.683200	
std	2.702377	2.68117	2.471258	2.946626	3.085511	
min	0.000000	0.00000	0.000000	0.000000	0.000000	
25%	1.900000	1.20000	4.700000	1.300000	1.100000	
50%	4.000000	2.90000	6.700000	3.200000	2.500000	
75%	6.125000	5.40000	8.300000	6.000000	6.100000	
max	10.000000	10.00000	10.000000	10.000000	10.000000	

	Writing	...	Art	Shopping	\
count	10000.000000	...	10000.000000	10000.000000	
mean	2.783220	...	3.694960	5.537580	
std	2.641601	...	2.597541	2.626148	
min	0.000000	...	0.000000	0.000000	
25%	0.875000	...	1.500000	3.500000	
50%	1.700000	...	3.300000	5.600000	
75%	4.200000	...	5.600000	7.700000	
max	10.000000	...	10.000000	10.000000	

	Social Media	Reading	Socializing	Gaming	\
count	10000.000000	10000.000000	10000.000000	10000.000000	
mean	5.471920	5.067150	8.144480	5.663800	
std	2.617237	2.725082	1.537818	3.150193	
min	0.000000	0.000000	2.000000	0.000000	
25%	3.500000	2.900000	7.400000	2.700000	
50%	5.500000	5.000000	8.600000	6.100000	
75%	7.600000	7.300000	9.300000	8.600000	
max	10.000000	10.000000	10.000000	10.000000	

	Entertainment Spending	Clothing Spending	Internet Spending	\
count	10000.000000	10000.000000	10000.000000	
mean	5.422330	5.240530	4.785600	
std	2.438987	2.460029	2.631747	
min	0.000000	0.000000	0.000000	
25%	3.700000	3.400000	2.700000	
50%	5.400000	5.300000	4.700000	
75%	7.300000	7.100000	6.800000	

```
max                  10.000000          10.000000          10.000000

       Retail Spending
count      10000.000000
mean           6.113950
std            2.254891
min            0.000000
25%            4.600000
50%            6.300000
75%            7.800000
max           10.000000

[8 rows x 22 columns]
```

In [8]:
```
df.corr()
```

Out[8]:
```
                             Age     Music  Movies/Theaters  Tech/Gadgets  \
Age                     1.000000 -0.071404        -0.032861      0.049297
Music                  -0.071404  1.000000         0.168541     -0.002268
Movies/Theaters        -0.032861  0.168541         1.000000      0.054879
Tech/Gadgets            0.049297 -0.002268         0.054879      1.000000
Museums                 0.023706  0.044854         0.054395     -0.092826
Food/Dining             0.034932 -0.007305        -0.021373     -0.054901
Camping/Hiking          0.064568  0.080073         0.054241      0.005400
Concerts               -0.002136  0.107177         0.056767     -0.062512
Clubs/Dancing          -0.001543  0.101726        -0.034442     -0.051267
Writing                 0.032917  0.034163        -0.007181     -0.168813
Sports                  0.010770  0.011773         0.035426      0.235932
Gardening               0.087740 -0.002706        -0.004820     -0.017819
Art                    -0.031547  0.041166         0.102754      0.017955
Shopping               -0.123566  0.103254         0.121079     -0.018813
Social Media            0.097224  0.028320         0.067204      0.307223
Reading                 0.037083  0.085666         0.082228     -0.173863
Socializing            -0.076566  0.197688         0.155671      0.069157
Gaming                 -0.043938  0.074207         0.108316      0.016821
Entertainment Spending -0.043698  0.045703         0.071656      0.130339
Clothing Spending      -0.079546  0.092160         0.065289      0.072743
Internet Spending      -0.012542  0.004195         0.106014      0.293165
Retail Spending         0.004743  0.058070         0.010685      0.059534

                         Museums  Food/Dining  Camping/Hiking  Concerts  \
Age                     0.023706     0.034932        0.064568 -0.002136
Music                   0.044854    -0.007305        0.080073  0.107177
Movies/Theaters         0.054395    -0.021373        0.054241  0.056767
Tech/Gadgets           -0.092826    -0.054901        0.005400 -0.062512
Museums                 1.000000     0.246781        0.245504  0.231917
Food/Dining             0.246781     1.000000        0.217308  0.147504
Camping/Hiking          0.245504     0.217308        1.000000  0.226031
```

Concerts	0.231917	0.147504	0.226031	1.000000
Clubs/Dancing	0.282887	0.251260	0.216383	0.215670
Writing	0.336356	0.194001	0.066437	0.153924
Sports	0.000237	-0.021753	0.092645	0.221547
Gardening	0.213505	0.145903	0.242661	0.217748
Art	0.032338	-0.071957	-0.020716	0.149597
Shopping	0.114030	-0.012112	0.032631	0.250630
Social Media	0.021101	0.071891	0.054334	-0.053998
Reading	0.541309	0.180555	0.211739	0.259666
Socializing	0.070987	-0.042491	0.054368	0.144199
Gaming	0.054911	-0.041373	0.109088	0.091409
Entertainment Spending	0.026435	-0.083160	-0.070372	0.007267
Clothing Spending	0.051895	-0.059216	-0.056388	0.158383
Internet Spending	-0.041650	-0.060884	-0.048324	-0.005129
Retail Spending	0.094143	-0.026182	0.087006	0.079134

	Clubs/Dancing	Writing	...	Art	\
Age	-0.001543	0.032917	...	-0.031547	
Music	0.101726	0.034163	...	0.041166	
Movies/Theaters	-0.034442	-0.007181	...	0.102754	
Tech/Gadgets	-0.051267	-0.168813	...	0.017955	
Museums	0.282887	0.336356	...	0.032338	
Food/Dining	0.251260	0.194001	...	-0.071957	
Camping/Hiking	0.216383	0.066437	...	-0.020716	
Concerts	0.215670	0.153924	...	0.149597	
Clubs/Dancing	1.000000	0.316432	...	-0.051388	
Writing	0.316432	1.000000	...	0.018504	
Sports	0.033086	-0.001682	...	0.029310	
Gardening	0.163847	0.211385	...	0.132446	
Art	-0.051388	0.018504	...	1.000000	
Shopping	-0.056642	0.001256	...	0.459960	
Social Media	0.041472	0.045665	...	-0.137035	
Reading	0.208205	0.247306	...	0.027576	
Socializing	0.039980	-0.038086	...	0.066582	
Gaming	-0.011205	0.005247	...	0.137589	
Entertainment Spending	-0.022475	-0.053478	...	0.036590	
Clothing Spending	-0.033141	0.003836	...	0.299727	
Internet Spending	-0.046308	-0.044473	...	0.057818	
Retail Spending	-0.025401	0.034375	...	0.027884	

	Shopping	Social Media	Reading	Socializing	\
Age	-0.123566	0.097224	0.037083	-0.076566	
Music	0.103254	0.028320	0.085666	0.197688	
Movies/Theaters	0.121079	0.067204	0.082228	0.155671	
Tech/Gadgets	-0.018813	0.307223	-0.173863	0.069157	
Museums	0.114030	0.021101	0.541309	0.070987	
Food/Dining	-0.012112	0.071891	0.180555	-0.042491	
Camping/Hiking	0.032631	0.054334	0.211739	0.054368	

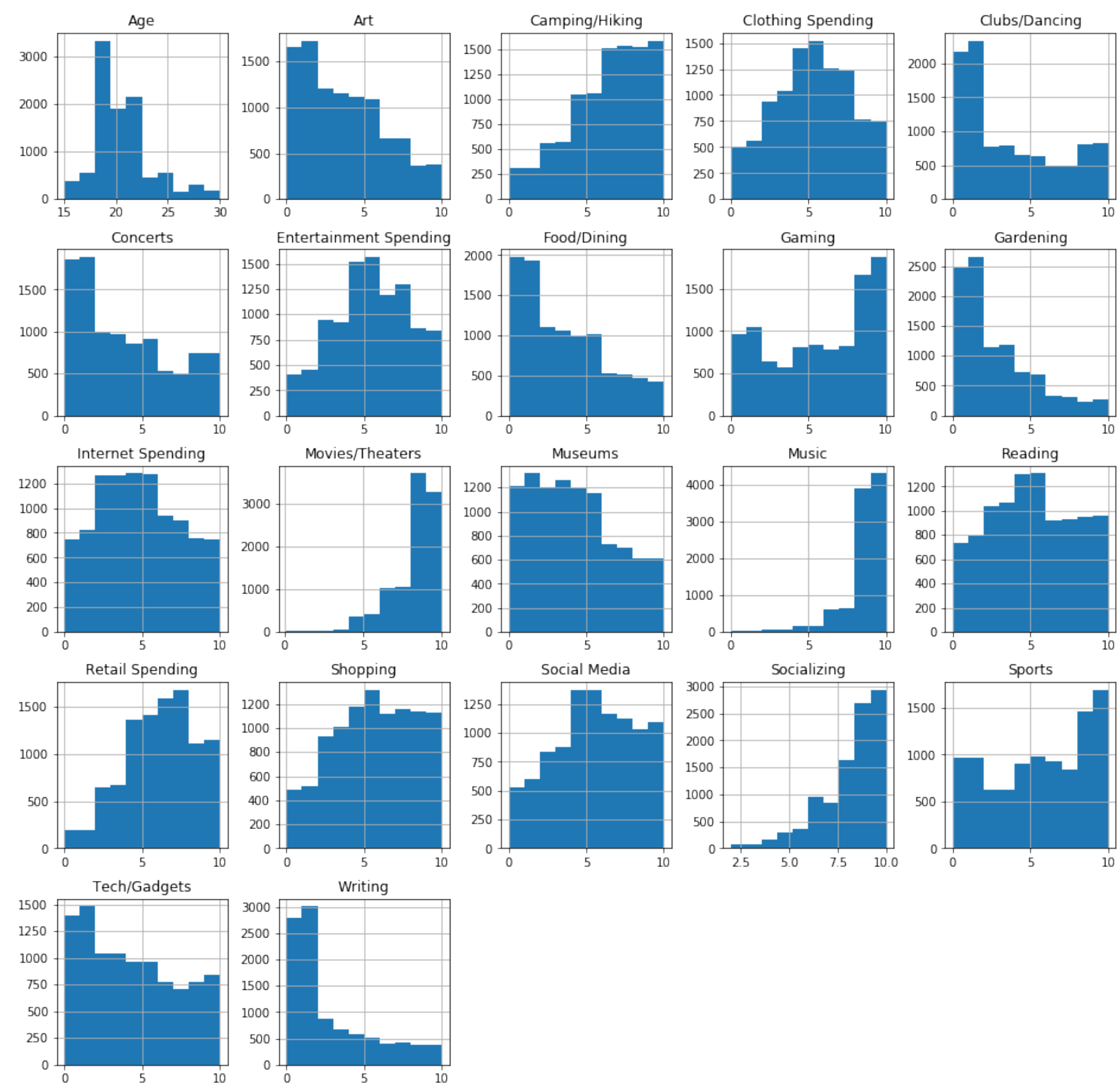
Concerts	0.250630	-0.053998	0.259666	0.144199
Clubs/Dancing	-0.056642	0.041472	0.208205	0.039980
Writing	0.001256	0.045665	0.247306	-0.038086
Sports	0.033294	0.136206	-0.051884	0.103949
Gardening	0.142955	0.035325	0.156246	0.021505
Art	0.459960	-0.137035	0.027576	0.066582
Shopping	1.000000	-0.107422	0.147850	0.173518
Social Media	-0.107422	1.000000	0.012539	0.018387
Reading	0.147850	0.012539	1.000000	0.145601
Socializing	0.173518	0.018387	0.145601	1.000000
Gaming	0.178581	0.030771	0.092676	0.062998
Entertainment Spending	0.035679	0.089676	-0.007857	0.242676
Clothing Spending	0.492331	-0.028645	0.055946	0.156786
Internet Spending	0.026846	0.278047	-0.039291	0.091446
Retail Spending	0.097629	0.122667	0.080410	0.128596

	Gaming	Entertainment Spending	Clothing Spending \
Age	-0.043938	-0.043698	-0.079546
Music	0.074207	0.045703	0.092160
Movies/Theaters	0.108316	0.071656	0.065289
Tech/Gadgets	0.016821	0.130339	0.072743
Museums	0.054911	0.026435	0.051895
Food/Dining	-0.041373	-0.083160	-0.059216
Camping/Hiking	0.109088	-0.070372	-0.056388
Concerts	0.091409	0.007267	0.158383
Clubs/Dancing	-0.011205	-0.022475	-0.033141
Writing	0.005247	-0.053478	0.003836
Sports	0.052906	0.103712	0.095491
Gardening	0.154157	-0.114303	-0.009075
Art	0.137589	0.036590	0.299727
Shopping	0.178581	0.035679	0.492331
Social Media	0.030771	0.089676	-0.028645
Reading	0.092676	-0.007857	0.055946
Socializing	0.062998	0.242676	0.156786
Gaming	1.000000	0.008179	0.078676
Entertainment Spending	0.008179	1.000000	0.369864
Clothing Spending	0.078676	0.369864	1.000000
Internet Spending	0.055776	0.336206	0.324058
Retail Spending	0.033115	0.129498	0.226647

	Internet Spending	Retail Spending
Age	-0.012542	0.004743
Music	0.004195	0.058070
Movies/Theaters	0.106014	0.010685
Tech/Gadgets	0.293165	0.059534
Museums	-0.041650	0.094143
Food/Dining	-0.060884	-0.026182
Camping/Hiking	-0.048324	0.087006

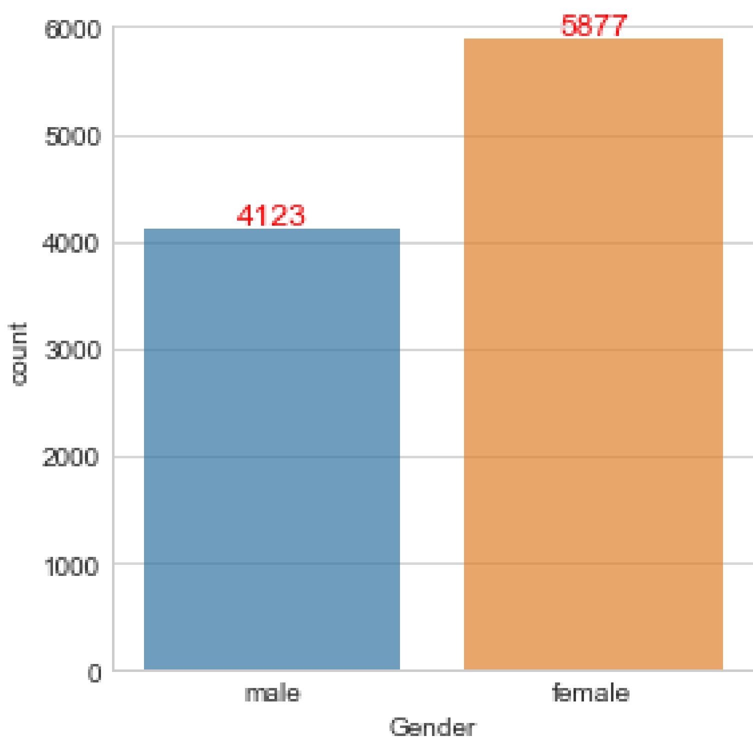
```
Concerts                          -0.005129          0.079134
Clubs/Dancing                     -0.046308         -0.025401
Writing                           -0.044473          0.034375
Sports                             0.136330          0.144912
Gardening                         -0.055597          0.014816
Art                                0.057818          0.027884
Shopping                           0.026846          0.097629
Social Media                       0.278047          0.122667
Reading                           -0.039291          0.080410
Socializing                        0.091446          0.128596
Gaming                             0.055776          0.033115
Entertainment Spending             0.336206          0.129498
Clothing Spending                  0.324058          0.226647
Internet Spending                  1.000000          0.212686
Retail Spending                    0.212686          1.000000

[22 rows x 22 columns]
```

In [11]:
```
df.hist(figsize=(16,16))
plt.show()
```



In [14]:
```
import math
# Set plotting style
sns.set_style('whitegrid')

# Rounding the integer to the next hundredth value plus an offset of 100
def roundup(x):
    return 100 + int(math.ceil(x / 100.0)) * 100


sns.factorplot('Gender', data=df, kind='count', alpha=0.7, size=4, aspect=1)

# Get current axis on current figure
ax = plt.gca()
```

```
# ylim max value to be set
y_max = df['Gender'].value_counts().max()
ax.set_ylim([0, roundup(y_max)])

# Iterate through the list of axes' patches
for p in ax.patches:
    ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
            fontsize=12, color='red', ha='center', va='bottom')

plt.show()
```



###
Data Wrangling

Fortunately, the data provided is pretty clean. All numerical categories except gender, which we will change in cells below.

We are also interested in customers that spend on the internet as well as clothing.

```
In [15]: #changing categorical column to numerical
         df['Gender'] = df['Gender'].map({'female': 1, 'male': 0})

In [16]: df = df[(df['Clothing Spending'] > 5) & (df['Internet Spending'] > 5)]

In [17]: df.head()
```

```
Out[17]:       Age  Gender  Music  Movies/Theaters  Tech/Gadgets  Museums  Food/Dining  \
         0   17.0       0    7.3              8.1           2.8      1.6          4.5
         4   19.0       1    9.1              9.8           1.4      3.9          3.1
         18  20.0       1    9.7              9.4           5.1      4.4          0.7
         20  19.0       0    7.4              9.8           6.4      4.6          3.7
         21  19.0       0    6.6              8.5           9.5      7.5          0.4

             Camping/Hiking  Concerts  Clubs/Dancing        ...         Art  Shopping  \
         0              7.1       0.3            0.6        ...         2.9       9.9
         4              5.4       8.2            4.7        ...         4.5       4.4
         18             5.8       0.7            1.7        ...         5.0      10.0
         20             5.8       1.5            1.0        ...         4.9       7.2
         21             9.4       8.6            5.4        ...         4.8       5.6

             Social Media  Reading  Socializing  Gaming  Entertainment Spending  \
         0            4.4      1.4          8.7     6.7                     9.7
         4            5.1      5.0          6.8     3.0                     2.0
         18           3.6      4.8          9.4     8.5                     7.4
         20           0.3      4.6          7.7     7.3                     4.8
         21           7.5      7.9          2.2     1.7                     8.6

             Clothing Spending  Internet Spending  Retail Spending
         0                 8.6                9.5              8.2
         4                 5.9                5.6              8.2
         18                7.2                6.2              6.4
         20                5.3                5.7              5.2
         21                5.1                5.4              4.2

         [5 rows x 23 columns]
```

###
Machine Learning

A big part of machine learning is to decide which algorithm and why.

For any type of segmenting problem, a clustering algorithm works really well. It detects certain behaviors and patterns and its able to separate them into clusters. This in fact is exactly what we need, a market segmentation solution.

In the real world, you would want to try out a couple of clustering algorithms, test their performance, and ultimately choose the best one. For this project we will try K-means clustering. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

A well known supplement for k-means is called Principal Component Analysis which helps emphasize variation and bring out strong patterns in a dataset. It's often used to make data easy to explore and visualize. K-means and PCA are usually thought of as two very different problems: one as an algorithm for data clustering, and the other as a framework for data dimension reduction, but combined together they work really well.

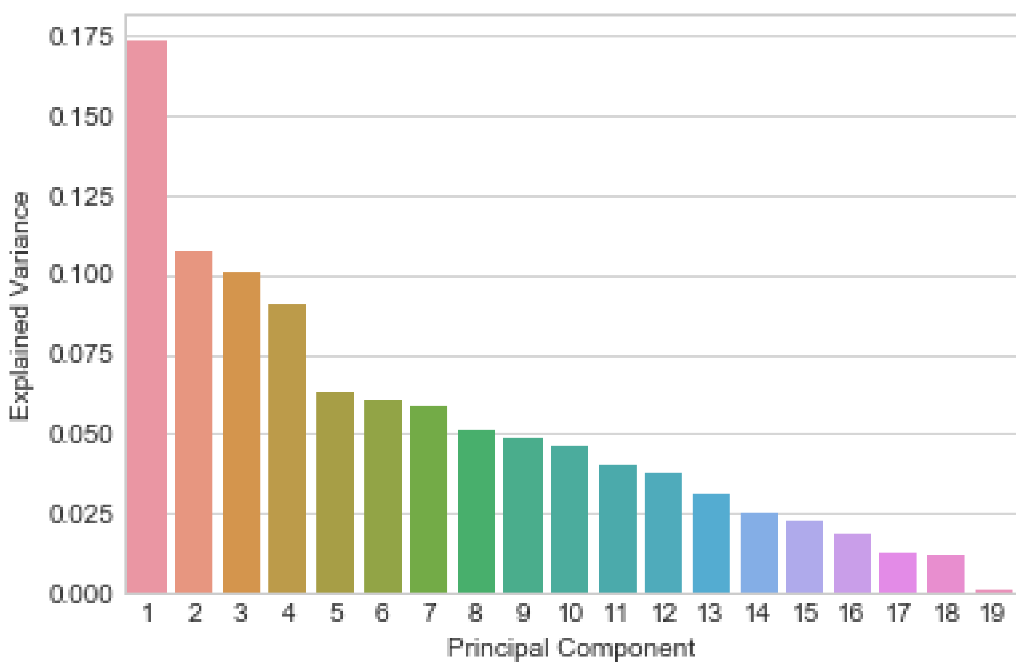
```
In [18]: # eliminating for now, variables that represent behaviors
         abt = df.drop(['Entertainment Spending',
                        'Clothing Spending',
                        'Internet Spending',
                        'Retail Spending'], axis=1)
```

```
In [19]: from sklearn.decomposition import PCA
```

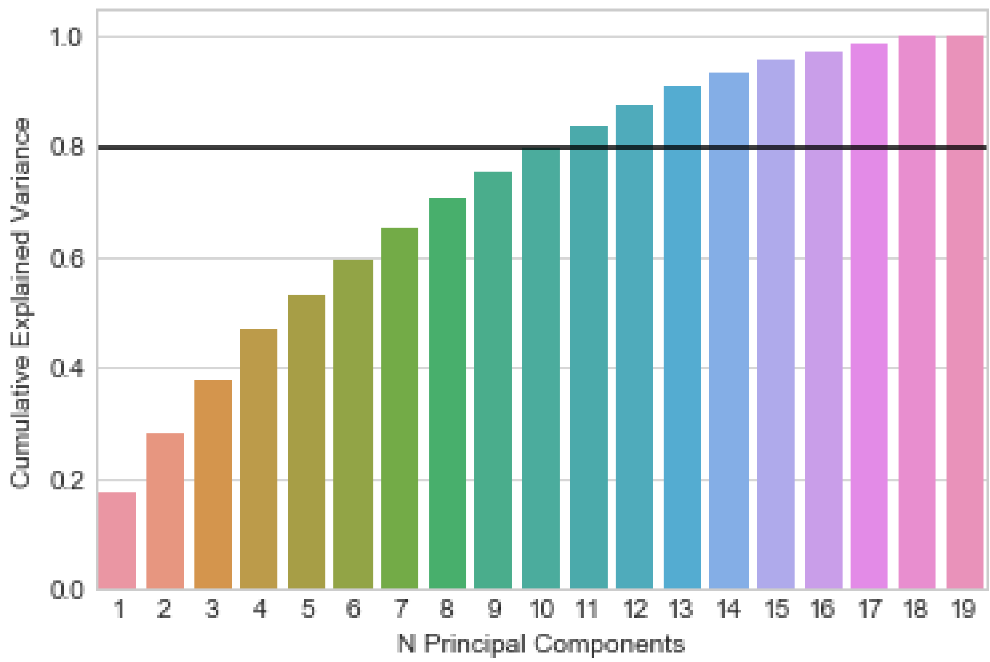
```
In [20]: pca = PCA()
         pca.fit( abt )
```

```
Out[20]: PCA(copy=True, iterated_power='auto', n_components=None, random_state=None,
           svd_solver='auto', tol=0.0, whiten=False)
```

```
In [21]: # Explained Variance Ratio
         sns.barplot(x=np.arange(pca.n_components_) + 1,
                     y=pca.explained_variance_ratio_)
         plt.xlabel('Principal Component')
         plt.ylabel('Explained Variance')
         plt.show()
```



In [22]:
```
# Cumulative Explained Variance
cumulative = np.cumsum(pca.explained_variance_ratio_)
sns.barplot(x=np.arange(pca.n_components_) + 1,
            y=cumulative)
plt.xlabel('N Principal Components')
plt.ylabel('Cumulative Explained Variance')
plt.axhline(y=0.8, color='k', linestyle='-')
plt.show()
```



Capture atleast 80% of the variance

In [23]:
```
from sklearn.cluster import KMeans
```

In [24]:
```
# PCA transformation
pc_df = pd.DataFrame( pca.transform(abt) )

# Rename Columns
pc_df.columns = ['PC{}'.format(n+1) for n in np.arange(pca.n_components_)]

pc_df.head()
```

Out[24]:

```
        PC1       PC2       PC3       PC4       PC5       PC6       PC7  \
0 -4.718882 -2.960997  1.116481  2.548510 -0.199757 -1.365595 -0.791734
1  0.491975  1.380983  1.128373  6.301044 -0.359154  0.709515 -1.014473
2 -2.770408 -4.774232  1.598379  0.078869 -0.386427 -0.448358 -1.733045
```

```
3 -3.041096 -3.171436  0.658393  2.253144  0.320050 -0.796833 -2.609572
4  2.375304  1.511895 -0.941456 -0.373477  1.079997 -4.262592  0.463675

        PC8       PC9      PC10      PC11      PC12      PC13      PC14  \
0 -5.314410  2.321042 -2.127782  2.549628 -0.703628  0.477296 -2.276950
1  1.097250  1.166798  0.156752  1.311644  1.418522 -0.526075  0.050525
2 -1.781863 -2.426829 -2.196854  1.731944 -2.267540 -1.811011 -1.876973
3 -2.273643 -1.241404 -5.576016 -1.677364 -1.871108 -0.712583 -0.352017
4  7.823742  0.350148  1.699506 -1.973704 -1.237663  0.716793 -0.425342

       PC15      PC16      PC17      PC18      PC19
0  2.987285  1.136840  0.809147  0.100033  0.606076
1 -2.956461  0.213495 -0.668280 -1.945622 -0.374975
2  1.532407 -1.276738 -0.521122 -0.068454 -0.291319
3  0.611251  0.482353  0.841736 -1.471987  0.608281
4 -2.357721  5.414301 -1.630552 -4.333861  0.544460
```

In [25]:
```
# Create training set
pcs_to_keep = ['PC{}'.format(n+1) for n in np.arange(11)]
X_train = pc_df[pcs_to_keep]

# Train K-Means clustering algorithm
kmeans = KMeans(n_clusters=3)
kmeans.fit(X_train)
```

Out[25]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [26]:
```
def predict_clusters(raw_data, trained_pca, trained_kmeans, n_pc):
    df_new = raw_data.copy()

    # Filter to our target audience
    df_new = df_new[(df_new['Clothing Spending'] > 5)
                    & (df_new['Internet Spending'] > 5)]

    # Engineer Features
    df_new.rename(columns={'Gender':'Female'}, inplace=True)
    df_new.Female.replace({'female':1, 'male':0}, inplace=True)
    abt_new = df_new.drop(['Entertainment Spending',
                           'Clothing Spending',
                           'Internet Spending',
                           'Retail Spending'], axis=1)

    # PCA transformation
    pc_df_new = pd.DataFrame( trained_pca.transform(abt_new) )

    # Rename Columns
```

```
    pc_df_new.columns = ['PC{}'.format(n+1)
                         for n in np.arange(trained_pca.n_components_)]

    # Create test set
    pcs_to_keep = ['PC{}'.format(n+1) for n in np.arange(n_pc)]
    X_new = pc_df_new[pcs_to_keep]

    # Predict clusters
    df_new['Cluster'] = trained_kmeans.predict(X_new)

    return df_new[['Age', 'Female', 'Clothing Spending', 'Internet Spending', 'Cluste
```

In [27]:
```
raw_df = pd.read_csv('unseen_raw_data.csv')

raw_df
```

Out[27]:

	Age	Gender	Music	Movies/Theaters	Tech/Gadgets	Museums	Food/Dining	\
0	21.0	female	9.5	7.2	0.8	3.5	3.8	
1	23.0	female	9.1	8.6	3.3	5.5	3.8	
2	19.0	female	8.2	9.5	0.9	4.3	4.1	
3	19.0	female	8.8	6.3	0.1	3.9	3.2	
4	22.0	female	9.6	9.9	4.5	5.6	1.0	
5	19.0	male	9.0	7.0	5.6	5.2	6.4	
6	21.0	male	8.4	7.7	2.3	2.1	8.9	
7	20.0	male	9.7	8.5	6.8	4.7	0.4	
8	28.0	male	9.9	8.2	1.2	0.7	5.4	
9	19.0	female	10.0	9.3	3.8	9.8	4.3	
10	21.0	male	8.4	9.2	7.0	2.3	5.0	
11	20.0	male	9.9	9.8	6.9	6.4	1.0	
12	18.0	male	8.5	10.0	8.1	3.9	4.9	
13	17.0	male	8.1	8.6	6.5	5.0	1.0	
14	30.0	female	7.9	9.5	8.4	1.0	1.9	
15	16.0	female	9.8	8.6	0.5	0.5	1.8	
16	21.0	male	8.6	9.5	5.6	3.8	0.5	
17	18.0	female	8.3	9.0	3.0	6.1	0.9	
18	21.0	male	8.8	7.7	0.7	0.3	3.4	
19	23.0	male	9.0	8.3	8.9	2.5	1.7	
20	18.0	female	8.6	7.9	1.2	2.2	1.9	
21	28.0	female	8.9	4.1	5.1	5.7	8.0	
22	19.0	female	9.7	9.4	0.1	7.2	5.3	
23	17.0	female	9.1	9.3	6.8	9.8	6.5	
24	17.0	male	9.7	8.1	8.2	5.0	8.0	
25	19.0	male	10.0	8.4	9.0	1.6	1.9	
26	18.0	male	8.7	4.7	7.7	2.3	1.4	
27	20.0	female	8.2	9.5	1.7	9.2	0.9	
28	22.0	female	9.5	8.7	6.7	3.8	7.6	
29	19.0	male	8.3	7.1	9.4	3.0	4.9	
30	19.0	male	7.5	8.7	4.0	0.9	0.3	

31	21.0	female	9.9	9.0	5.5	9.4	4.0
32	17.0	female	8.1	8.5	1.5	2.1	1.5
33	30.0	male	8.9	9.7	7.7	9.4	10.0
34	18.0	female	9.7	7.2	8.9	0.6	0.4
35	19.0	female	6.3	6.6	1.9	7.0	9.9
36	20.0	female	9.5	9.6	3.3	7.7	2.9
37	20.0	male	9.6	9.8	9.0	0.4	0.9
38	20.0	female	9.4	9.5	6.4	3.3	3.0
39	29.0	male	6.3	9.7	5.5	1.0	0.4
40	19.0	male	6.1	9.6	4.7	1.6	1.1
41	18.0	female	9.3	1.7	8.4	9.1	9.1
42	17.0	female	8.5	8.8	4.6	0.6	3.9
43	20.0	male	10.0	7.2	8.4	0.4	1.8
44	25.0	female	9.1	8.8	0.8	9.0	0.6
45	19.0	female	8.1	8.5	0.3	1.3	0.6
46	21.0	male	9.4	8.1	7.6	5.4	6.3
47	26.0	male	9.3	9.6	3.3	4.8	8.7
48	23.0	female	5.8	6.8	7.4	2.5	4.2
49	17.0	female	8.3	10.0	8.0	6.6	3.4

	Camping/Hiking	Concerts	Clubs/Dancing	...	Art	Shopping	\
0	9.1	0.8	1.0	...	5.8	4.2	
1	6.5	6.1	2.5	...	5.4	4.2	
2	9.9	2.1	1.4	...	9.5	4.2	
3	9.4	9.5	1.9	...	1.9	5.1	
4	6.8	1.2	1.4	...	2.7	5.1	
5	9.6	8.7	8.1	...	0.7	1.8	
6	8.3	1.3	9.5	...	4.0	6.8	
7	2.8	1.9	0.7	...	4.5	5.4	
8	8.1	0.7	4.3	...	5.2	2.3	
9	6.5	6.1	0.3	...	4.4	6.2	
10	6.0	1.4	1.7	...	4.8	2.6	
11	7.5	4.6	1.9	...	3.4	5.1	
12	5.9	1.9	0.3	...	0.1	3.9	
13	4.1	1.0	0.3	...	2.2	5.7	
14	8.1	0.3	2.0	...	0.9	1.0	
15	8.8	0.5	1.5	...	3.9	9.8	
16	8.7	1.4	1.7	...	3.9	2.2	
17	8.2	3.4	8.4	...	0.3	8.5	
18	9.2	0.7	1.7	...	1.8	2.2	
19	6.7	0.7	3.8	...	0.8	2.0	
20	4.2	3.8	1.0	...	2.2	5.2	
21	8.0	4.0	7.5	...	7.0	7.5	
22	9.1	9.8	9.0	...	7.1	7.6	
23	7.8	3.4	7.8	...	0.8	3.5	
24	6.2	6.4	6.3	...	2.5	4.7	
25	7.6	0.2	0.1	...	7.9	9.4	
26	6.1	0.4	8.5	...	4.1	5.8	

27	8.9	5.5	1.7	...	5.8	10.0
28	9.9	5.7	2.0	...	6.8	7.2
29	9.5	2.1	0.3	...	0.1	3.5
30	8.8	2.9	1.6	...	0.1	1.0
31	8.1	8.1	0.1	...	8.2	8.3
32	6.6	2.5	1.7	...	2.8	4.9
33	9.4	7.9	7.2	...	4.4	5.9
34	6.8	0.9	1.0	...	7.5	5.2
35	9.7	1.4	9.3	...	2.0	3.4
36	9.8	7.8	7.1	...	1.4	1.8
37	6.7	0.2	0.2	...	0.4	6.0
38	6.6	2.4	1.1	...	5.1	5.4
39	3.7	0.0	2.7	...	0.0	5.2
40	0.9	1.1	4.6	...	0.5	2.3
41	8.9	9.6	8.7	...	1.5	8.4
42	7.7	5.7	0.6	...	8.4	9.0
43	4.4	1.8	1.9	...	1.2	2.1
44	9.8	7.2	0.7	...	4.4	8.9
45	4.6	4.6	1.5	...	8.4	8.9
46	9.0	5.9	8.6	...	8.1	8.7
47	9.1	7.1	5.8	...	1.7	5.3
48	9.9	0.3	1.4	...	7.0	6.8
49	7.1	8.0	4.7	...	1.4	9.8

	Social Media	Reading	Socializing	Gaming	Entertainment Spending	\
0	5.7	4.7	7.8	8.8	4.0	
1	6.7	3.1	8.0	8.4	4.8	
2	0.9	6.3	8.2	9.2	4.0	
3	0.0	2.4	8.4	0.9	3.7	
4	8.3	4.1	8.4	1.6	9.5	
5	9.8	6.2	8.3	1.1	6.0	
6	8.0	0.6	8.3	5.7	6.7	
7	5.4	3.6	7.7	5.5	5.5	
8	9.9	7.8	8.1	0.6	6.7	
9	4.9	9.8	9.3	8.3	5.7	
10	7.5	3.4	8.4	6.2	4.9	
11	1.8	1.8	6.2	8.4	4.8	
12	8.0	1.0	7.3	1.7	6.3	
13	3.2	0.9	8.9	8.7	8.6	
14	9.0	1.3	9.5	3.1	8.9	
15	1.1	1.7	8.0	8.9	4.4	
16	4.8	0.3	9.4	5.9	10.0	
17	9.1	8.1	9.8	9.0	8.1	
18	7.6	1.6	6.7	5.5	4.8	
19	9.6	2.8	9.1	1.1	5.9	
20	1.0	4.9	9.2	1.0	5.2	
21	6.8	4.2	7.8	5.5	8.2	
22	2.5	5.7	9.3	10.0	4.3	

23	5.5	6.3	8.4	7.2	1.5
24	8.9	9.5	8.1	5.7	9.0
25	0.1	0.7	8.8	0.5	5.7
26	6.3	3.0	9.6	0.5	6.9
27	1.6	9.8	9.4	9.5	8.2
28	5.7	1.0	6.6	8.7	6.8
29	5.4	4.7	9.9	0.4	5.8
30	4.5	2.5	9.3	0.2	5.1
31	5.8	10.0	8.9	9.4	4.2
32	6.2	6.4	6.4	1.9	3.5
33	6.2	9.4	6.6	8.0	6.2
34	5.5	3.4	2.6	8.1	0.3
35	5.8	6.1	7.4	1.4	5.1
36	2.1	6.8	10.0	0.6	4.5
37	9.0	0.3	8.0	1.0	5.3
38	6.6	5.7	7.8	9.1	4.0
39	9.3	2.9	7.6	8.0	4.2
40	6.5	1.0	7.5	7.3	7.1
41	1.8	8.6	9.0	0.6	0.3
42	4.6	3.2	8.7	7.1	7.4
43	6.2	2.5	6.4	1.8	5.5
44	9.5	8.2	9.7	7.7	8.5
45	1.2	6.2	9.2	0.3	8.8
46	8.5	9.1	8.5	9.3	6.5
47	7.2	6.6	9.6	1.3	8.4
48	6.8	6.5	8.1	9.0	6.8
49	7.8	6.3	8.1	4.6	7.1

	Clothing Spending	Internet Spending	Retail Spending
0	7.8	5.7	6.8
1	7.0	4.7	5.3
2	2.7	3.7	3.7
3	2.0	2.8	6.5
4	5.7	3.1	7.5
5	3.4	4.3	5.2
6	5.1	4.4	5.1
7	4.4	5.2	5.8
8	4.2	2.4	9.1
9	6.2	3.0	4.3
10	2.5	2.8	5.6
11	5.3	6.7	7.1
12	2.8	3.5	4.7
13	4.9	1.1	6.0
14	5.8	2.4	3.2
15	4.5	1.8	3.4
16	2.9	3.2	5.6
17	8.4	5.9	9.7
18	3.9	2.6	4.3

```
19              4.7            4.0           6.0
20              4.8            2.9           8.2
21              7.4            7.3           6.2
22              0.2            6.3           2.6
23              0.0            5.2           8.5
24              2.5            7.9           3.1
25              5.8            4.1           4.5
26              5.4            5.1           5.1
27              8.0            4.7           9.5
28              6.4            6.7           4.4
29              6.0            8.9           3.9
30              4.4            8.5           7.6
31              4.1            1.2           1.3
32              7.8            3.7           7.9
33              6.3            6.1           8.1
34              2.4            2.0           1.0
35              1.5            1.7           2.9
36              5.9            4.1           6.0
37              3.1            2.8           7.6
38              5.7            1.4           9.1
39              0.0            8.7           2.9
40              2.2            8.0           4.5
41              3.5            0.8           5.8
42              9.8            4.5           6.2
43              4.1            4.2           5.4
44              4.3            5.3           4.4
45              8.7            0.7           6.0
46              7.5            1.3           4.3
47              7.5            7.1           9.8
48              7.2            6.0           9.6
49              9.9            8.7           8.7

[50 rows x 23 columns]
```

In [29]:
```python
pred_df = predict_clusters(raw_df, pca, kmeans, 11)

pred_df
#THESE ARE THE CUSTOMERS THAT THE E-COMMERCE COMPANY NEEDS TO TARGET
```

Out[29]:

	Age	Female	Clothing Spending	Internet Spending	Cluster
0	21.0	1	7.8	5.7	1
11	20.0	0	5.3	6.7	1
17	18.0	1	8.4	5.9	1
21	28.0	1	7.4	7.3	0
26	18.0	0	5.4	5.1	2
28	22.0	1	6.4	6.7	0
29	19.0	0	6.0	8.9	2
33	30.0	0	6.3	6.1	0

47	26.0	0	7.5	7.1	0
48	23.0	1	7.2	6.0	1
49	17.0	1	9.9	8.7	0

###
Conclusion

In conclusion, we we're able tu succesfuly cluster those customers that have a high probability of spending online on clothing for the e-commerce company. As I said before, the ideal scenario would be to deploy 10+ models and observed their performance over some time. It would be incorrect to state that our algorithm is the "best", however it does accomplish the task with a certain level of success.