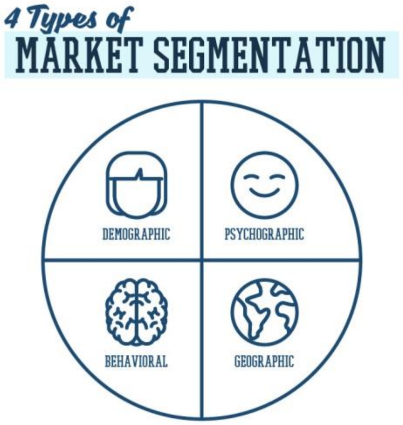
Application of Gaussian model in customer segmentation

Michael Meng, 1 Oct 2021

1 Project background and working environment setting

1.1 Project background introduction

Customer segmentation can help gain new customers by investing the company's resources on the right marketing channels with the right advertisement content. It will significantly increase the advertising effect while decreasing expenses. At the same time, the company can provide different after-sale services, discounts or other promotion methods to enlarge existing customers' potential.



Customers' grouping analysis is trying to find the patterns among customers various features and purchasing behaviors if they existed. Although we already have a general customer segmentation standard, we still need to specify or customize our segmentation method for different companies or particular company, even for specific promotion activity. Generally speaking, a specialized segmenting method is more effective than the generalized grouping standard.

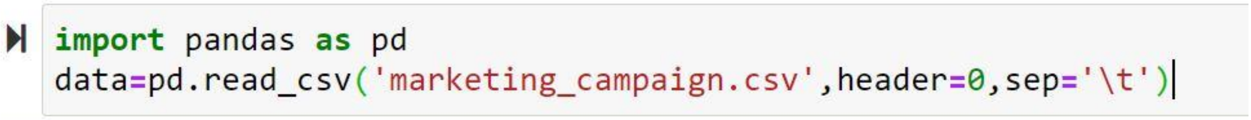
I picked a Kaggle website database and tried to verify the GaussianMixture model against specific customer segmentation analysis needs, and fortunately got a good result.

1.2 Data resources, license and summary

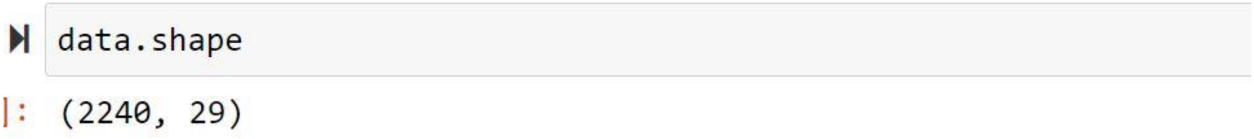
- The data was downloaded from Kaggle website(link)
- The data license is CC0: Public Domain(same link with previous line)
- The original data is a csv file with 2240 observations.
- Each record has 29 features describing detailed customers' personal information and historical purchasing and A/S service summaries.

1.3 Install required packages and other environments setting

The analysis will need Numpy and Pandas to interact with data, Pyplot and Seaborn libraries to plot and need GaussianMixture to analyze. And other Sklearn modules help to standardize or normalize data.



```
import pandas as pd
data=pd.read_csv('marketing_campaign.csv',header=0,sep='\t')
```



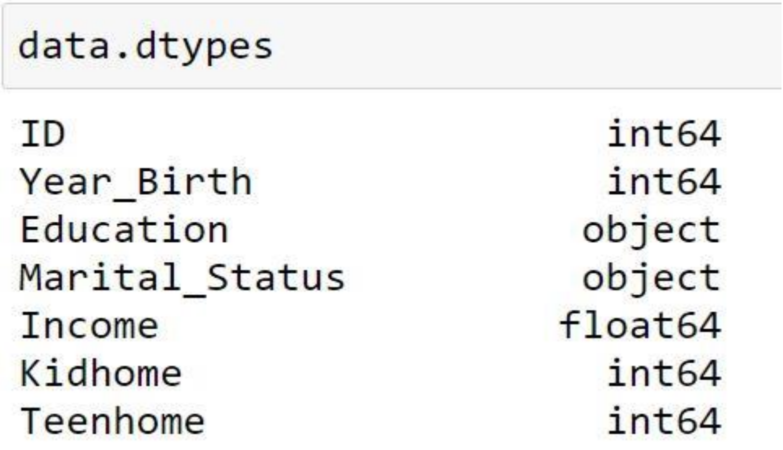
```
data.shape
```

```
]: (2240, 29)
```

2 Data inspecting and cleaning

2.1 Inspect data

At first, I need to get an overview of the dataset, for instance, the shape of the data, columns names and related meaning or definition, the relationship among different columns, and datatypes. Pandas has great functions or features to execute according to work.



```
data.dtypes
```

```
ID                  int64
Year_Birth          int64
Education          object
Marital_Status     object
Income            float64
Kidhome             int64
Teenhome            int64
```

Some features' datatypes were not aligned with their definition thus need to convert to suitable types.

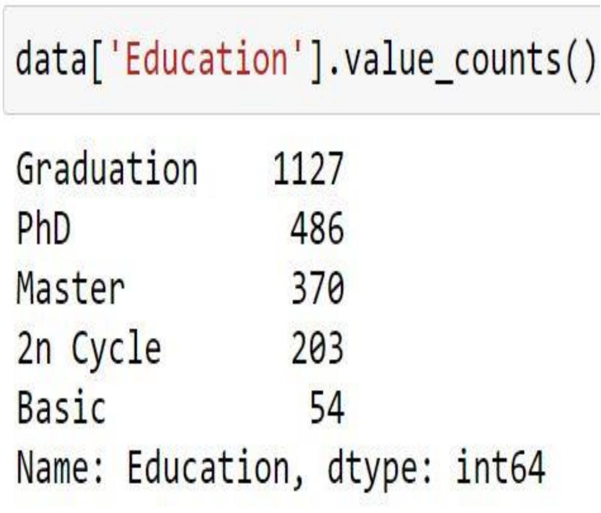
2.2 Missing value

Important feature (Income) contained null value, but 24 observations have the problem. I decided to delete the missing value instead of replacing them with mean or median, mode values.

2.3 The statistics of numeric data

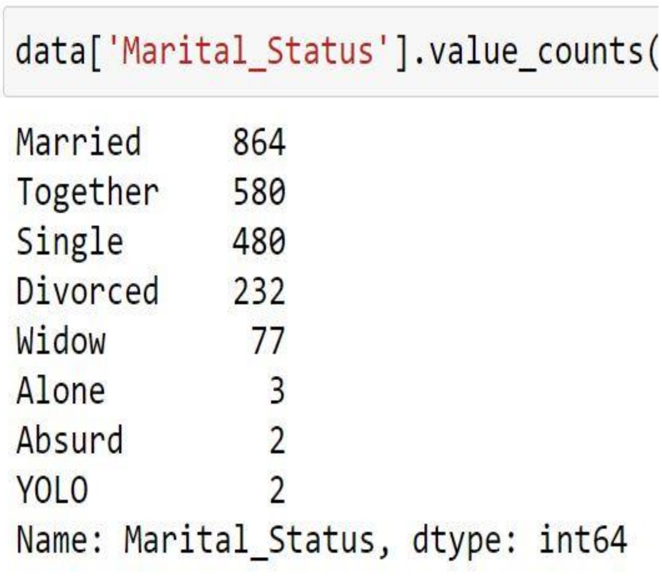
```
data.describe().T
```

2.4 View object features with unique value and its amount



```
data['Education'].value_counts()
```

```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```



```
data['Marital_Status'].value_counts(
```

```
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

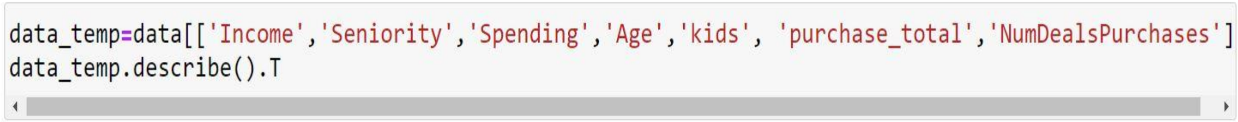
Education and Marital_Status columns were classified unclearly and arbitrarily, but I don't need the data on the following phrase. Otherwise, I need to replace the data with the proper format.

2.5 Clean data

3 Feature engineering

Because of overfitting problems, I need to deliver specific features to the model with a good scaler and not too many parts. I only got 2240 observations, so I had to pick up selected features and combine some features into one.

3.1 Select and combine features against ordinary domain knowledge



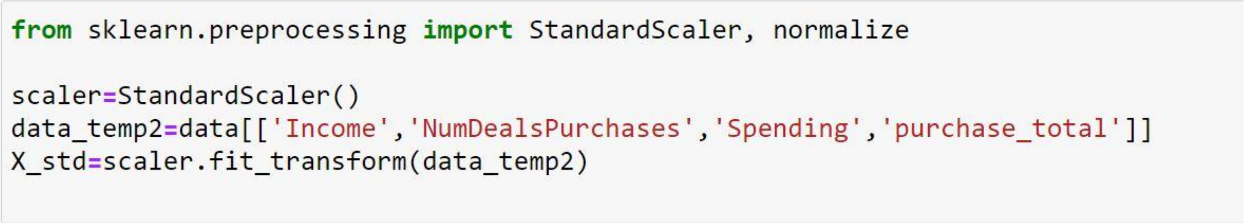
```
data_temp=data[['Income','Seniority','Spending','Age','kids', 'purchase_total','NumDealsPurchases']]
data_temp.describe().T
```

We may change our selection standard according to a different scenario. Black Friday marketing strategy's customer segmentation methods can't be the same as gaining new customers online. The following description is only one of the possible selections.

- Seniority is a variant of the existing column (Dt_Customer), representing the time since the customer's first Consumption.
- Spending is the summary of 6 category products, added six columns' value together.
- Purchase_total is also an addition of 3 columns; each represents a number of a particular consumption channel.
- Age, Kids are also some combination of other columns.
- NumDealsPurchases is an existing column more representative than the other six columns showing customers' attitudes towards promotion.

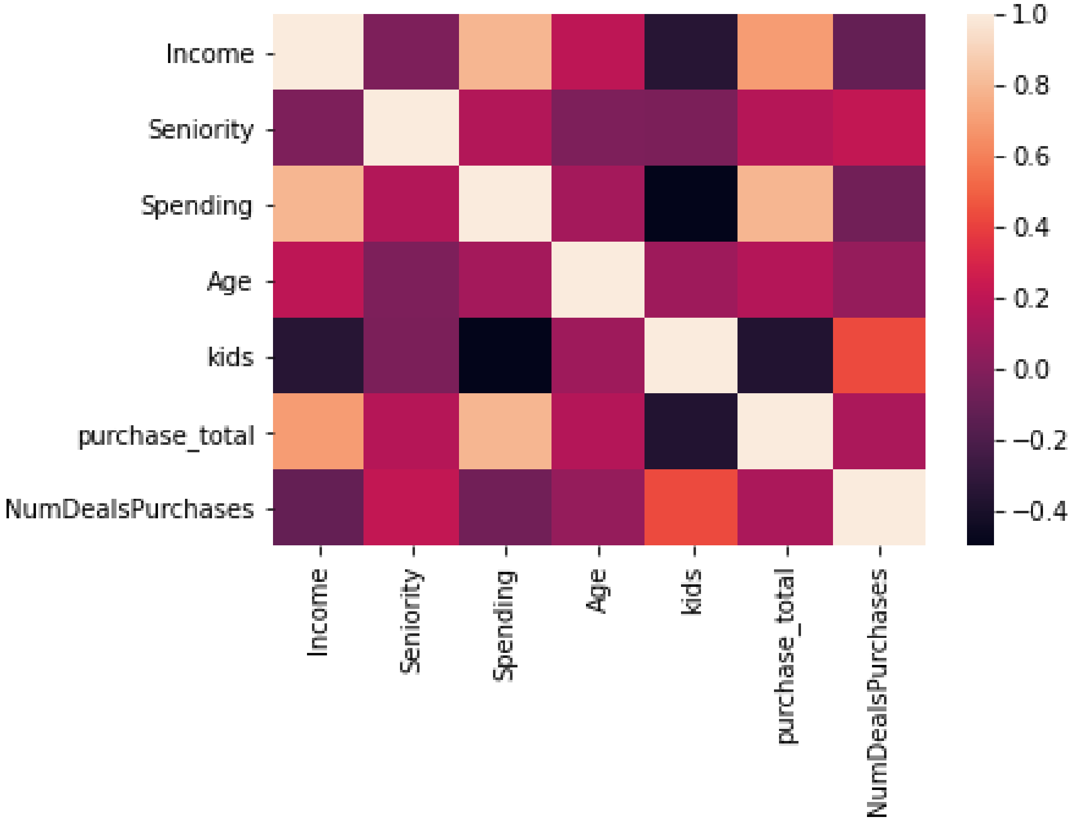
3.2 Feature scaler decision

After inspecting data minimum, maximum values and the distribution possibility, I will standardize income, spending, NumDealsPurchases and purchase_total features because those are more likely standardized distribution. And normalize Seniority and Age while passing kids numbers directly to the models.



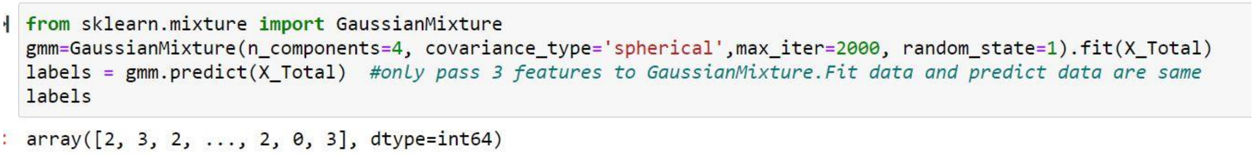
```
from sklearn.preprocessing import StandardScaler, normalize

scaler=StandardScaler()
data_temp2=data[['Income','NumDealsPurchases','Spending','purchase_total']]
X_std=scaler.fit_transform(data_temp2)
```

3.3 Combine the data and check the correlation among selected features



4 GaussianMixture model prediction and verification

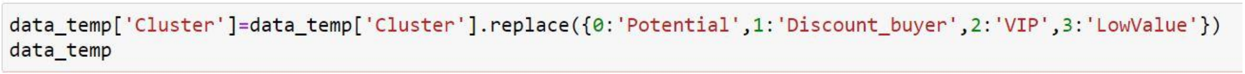
4.1 Get the results



```
from sklearn.mixture import GaussianMixture
gmm=GaussianMixture(n_components=4, covariance_type='spherical',max_iter=2000, random_state=1).fit(X_Total)
labels = gmm.predict(X_Total)  #only pass 3 features to GaussianMixture.Fit data and predict data are same
labels
```

```
array([2, 3, 2, ..., 2, 0, 3], dtype=int64)
```

After delivering the value to the Gaussian model, it is straightforward to get the results. For setting model components number is 4, we got an array with numbers 0-3. And we need to find the meaning of 0-3 and decide whether the classification can help us or not.



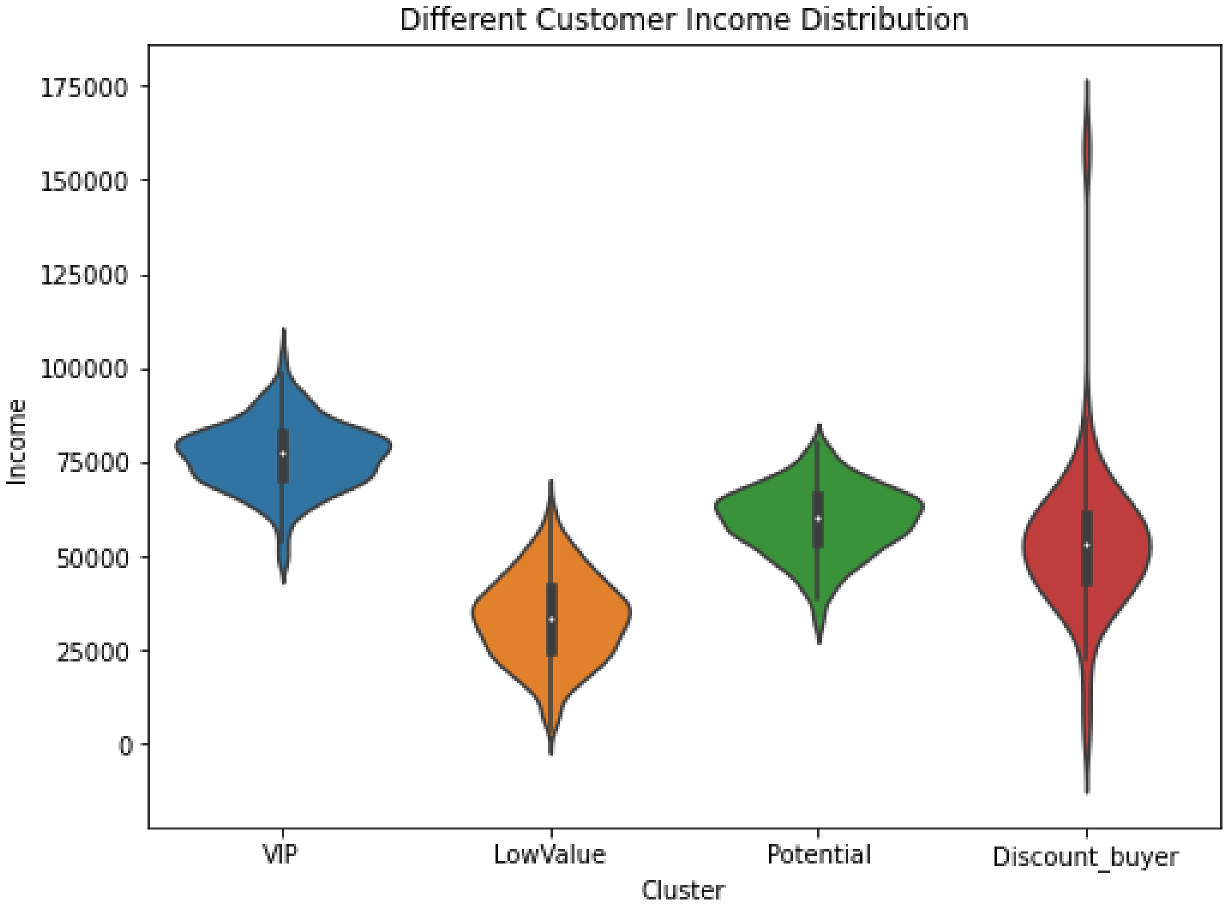
```
data_temp['Cluster']=data_temp['Cluster'].replace({0:'Potential',1:'Discount_buyer',2:'VIP',3:'LowValue'})
data_temp
```

The classification is meaningful and can help the company to do particular marketing strategies. The gaussian model knows the company's needs after training him with the proper transaction data.

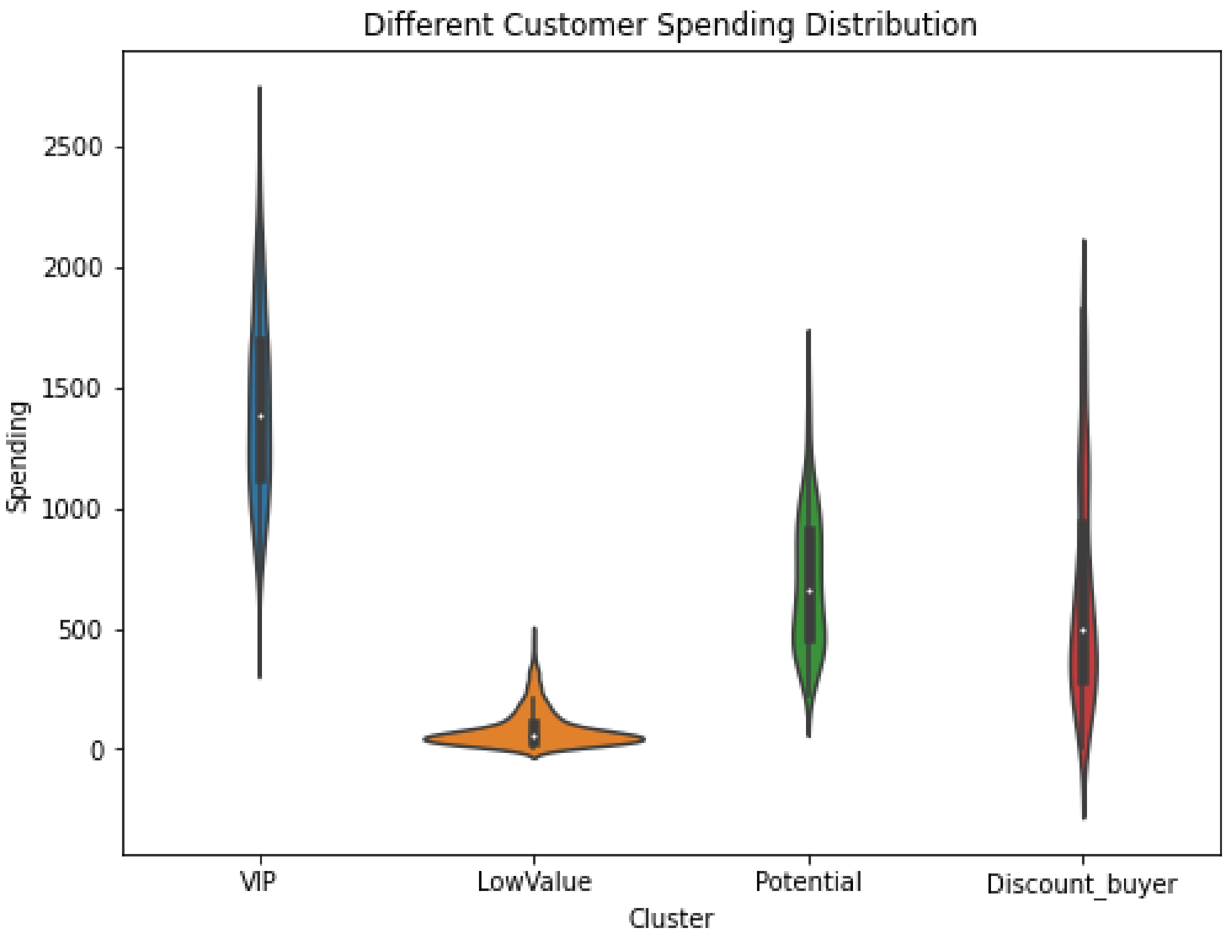
- 0：Potential customer
- 1: Discounts fans
- 2: The company’s VIP or star customer
- 3: Customers with low value

4.2 Single feature verification against customer types

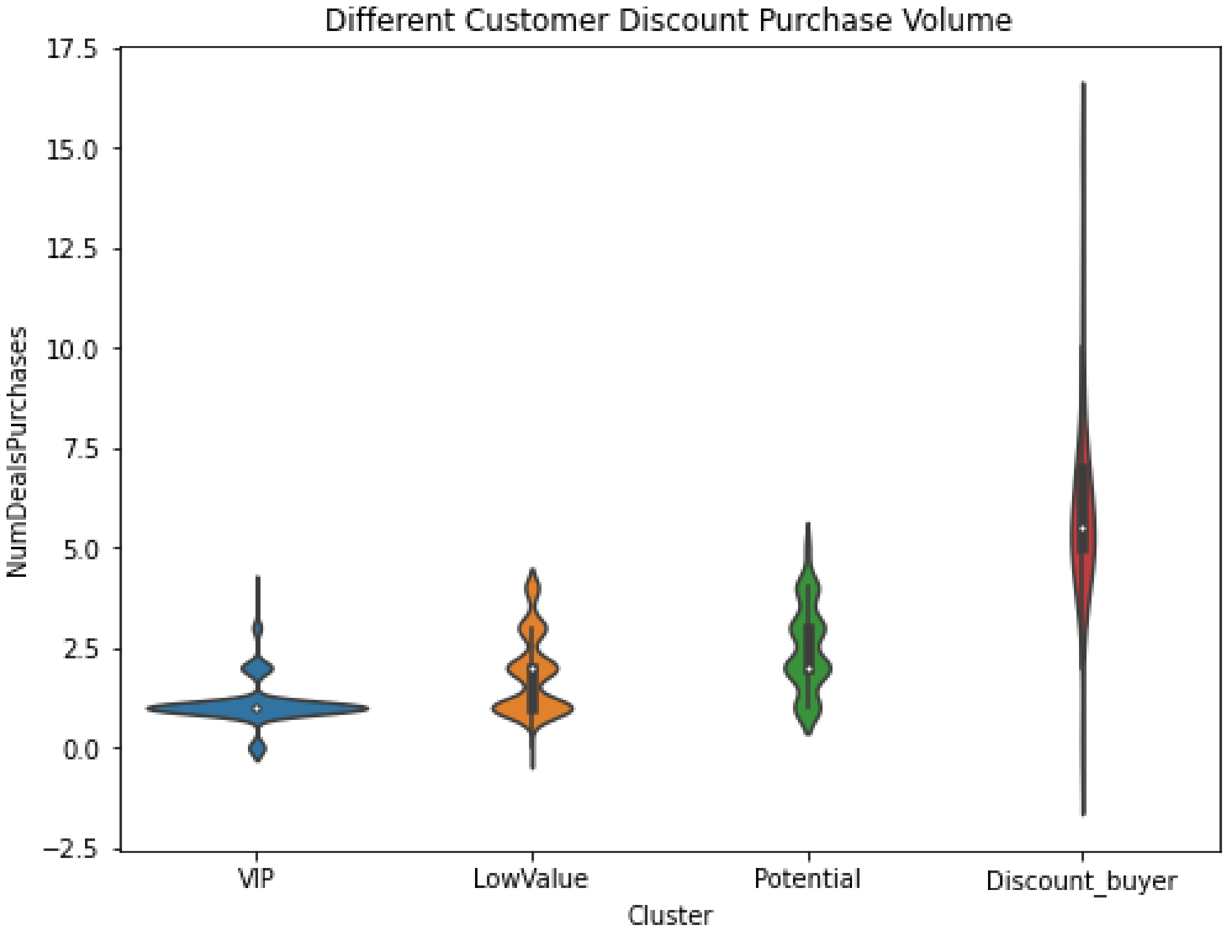
4.2.1Vip customers usually have good income against low value ones, but some discounts fans income distribution varies a lot.



4.2.2Spending difference

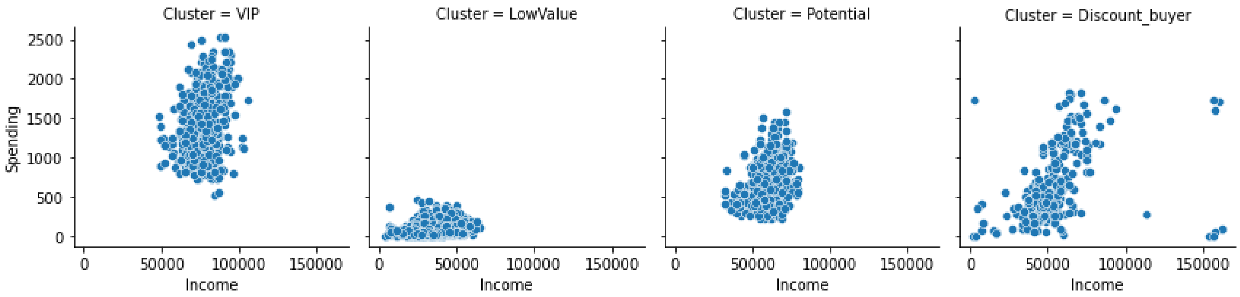


4.2.3 Price promotion acceptance numbers distribution among 4 type of customers

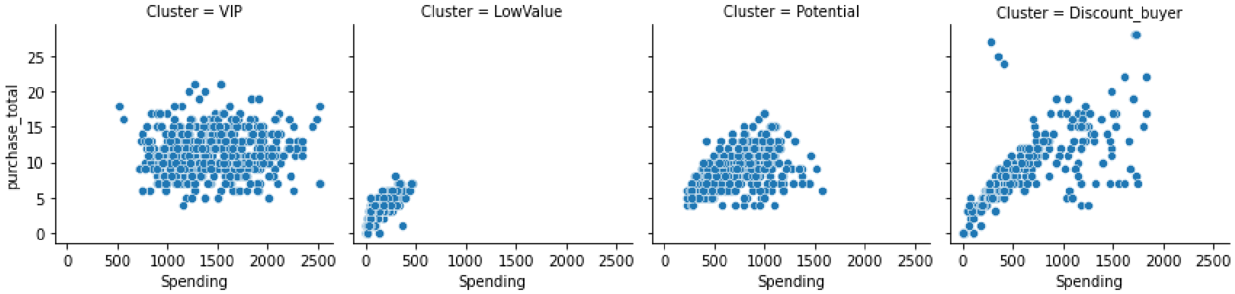


4.3 Dual variance analysis

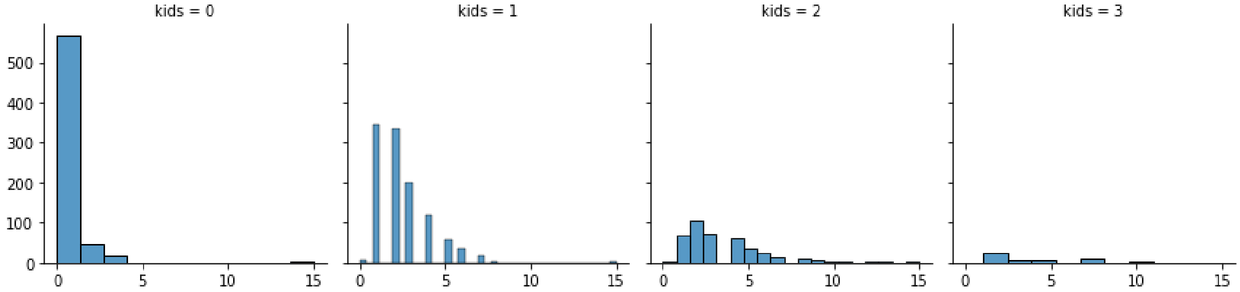
4.3.1Income and spending distribution among 4 type customers



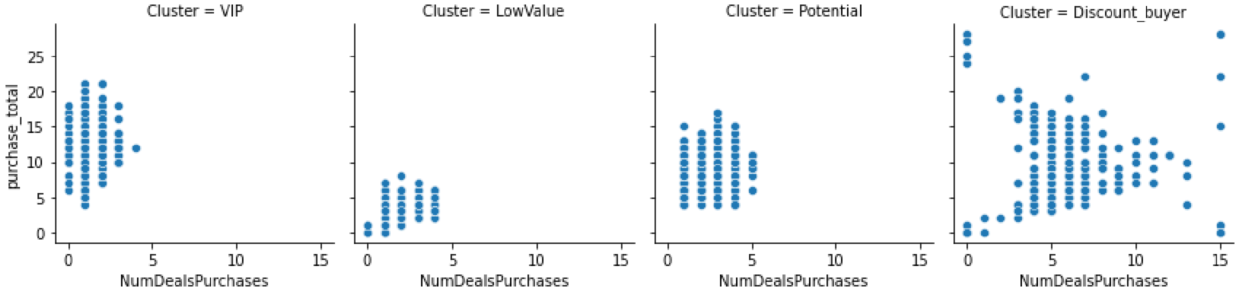
4.3.2Purchase_total and spending can show single consumption volume



4.3.3Family with kids usually be attracted by discounts



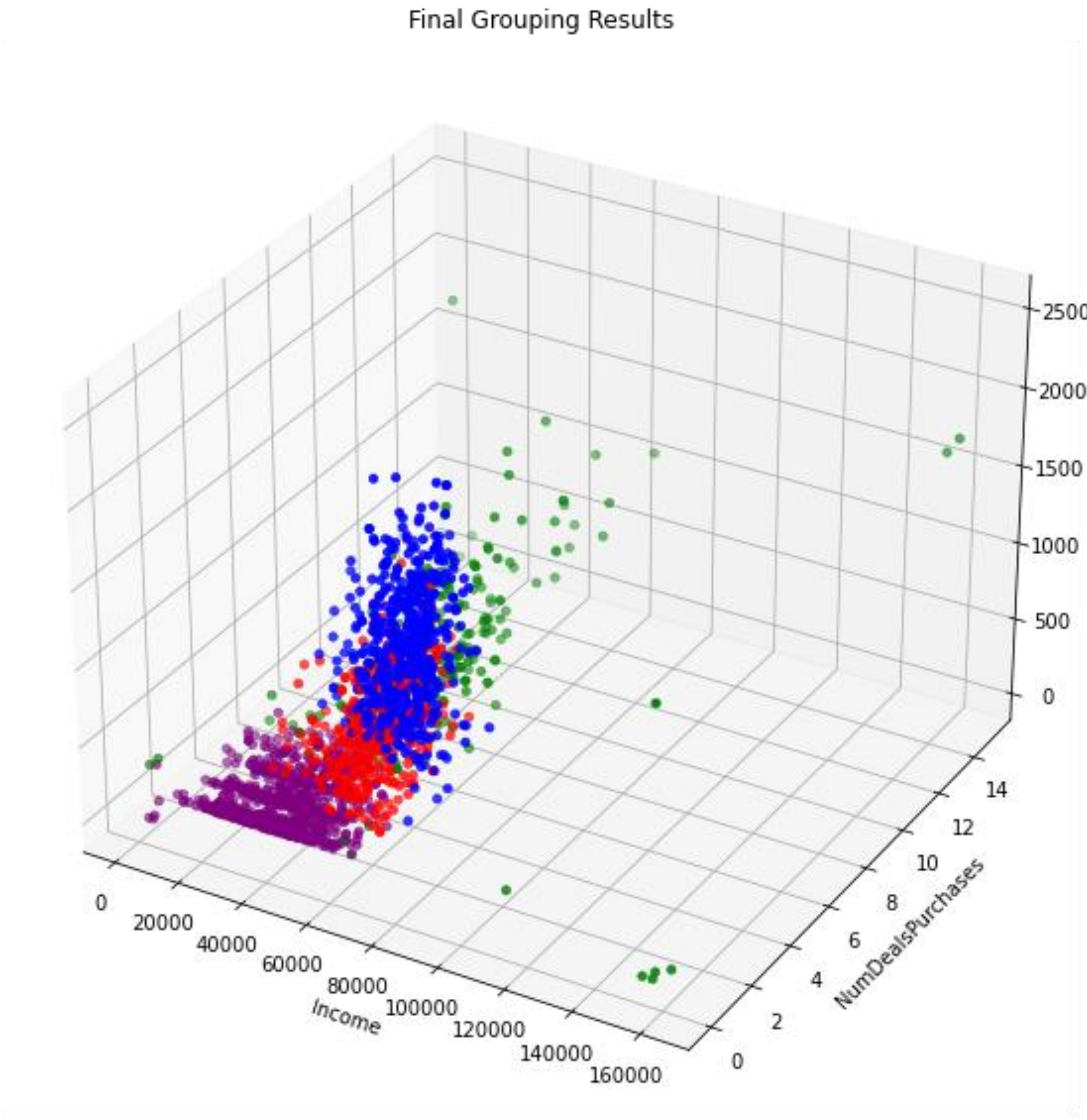
4.3.4The total number of consumption and discount consumption number



4.4 3D plot customer

I selected three more important features: income, historical spending, and promotion consumption number, to show the difference among customers. We can see the mean values and distribution among them.

Gaussian Model works this time, although diacount_buyer type is widely distributed, which need to fix in the future.



5 Conclusion and suggestion

5.1 Gaussian Model works for segment customers but need domain knowledge to justify the number of components and training data features.

5.2 Each marketing activity can own its segmentation.

5.3 The result of segmentation can be used to check the model. Supervised models are usually much powerful than unsupervised models.

6 Acknowledgments

The analysis process is based on Aman Kharwal's analysis (link). I am very grateful to Aman Kharwal, and I have learned a lot from him and his analysis.