Customer Personality Analysis

Clustering Problem :
The objective of the analysis is building to perform clustering to summarize customer segments.

Age | Marital Status | Education | Child | Income | Amount Spent | Total Accepted Campaign | Response

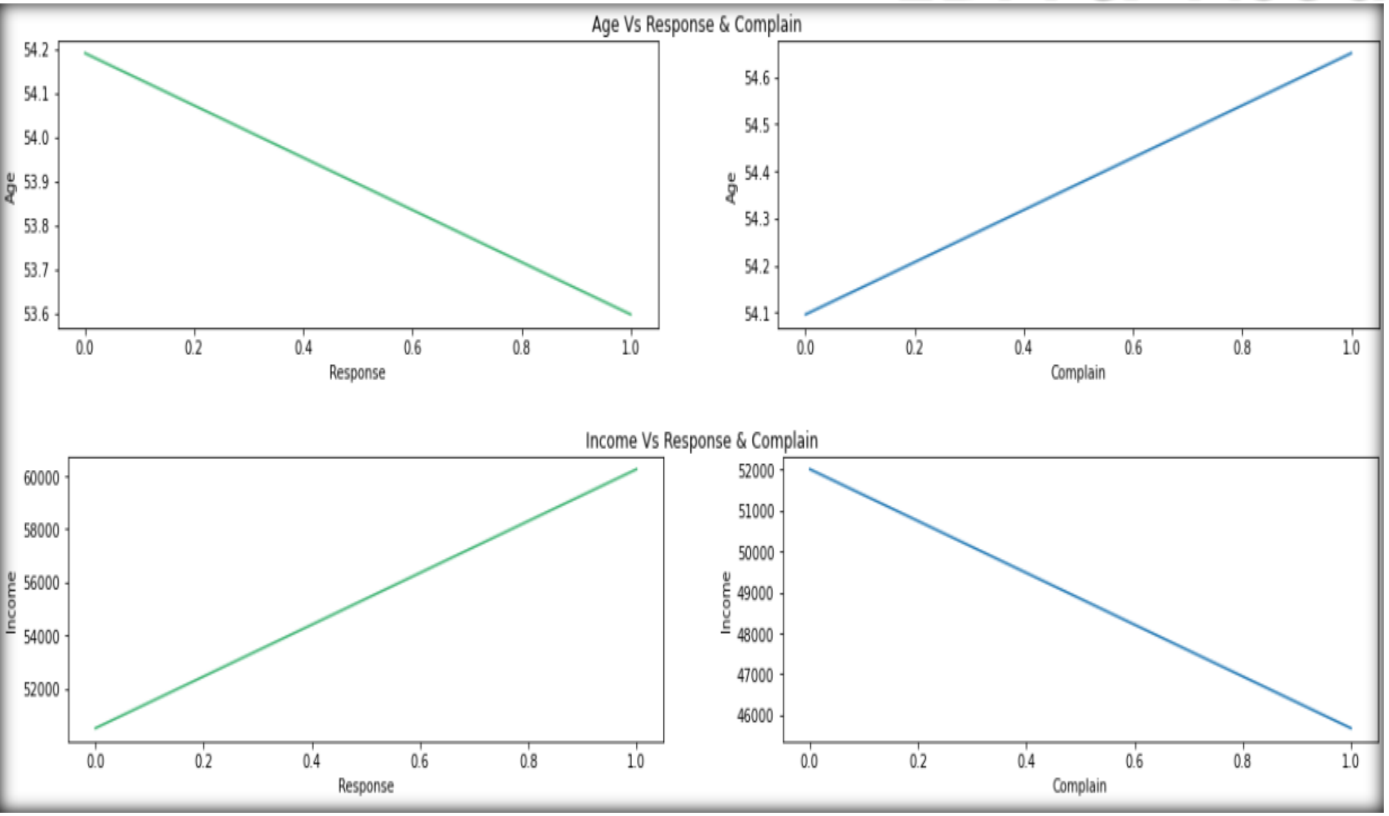
EDA & Visualization

Model Building

Clustering

Deployment

EDA & Visualization

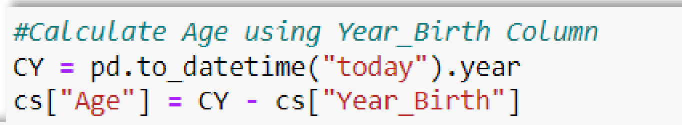


0 == NO RESPONSE NO COMPLAIN

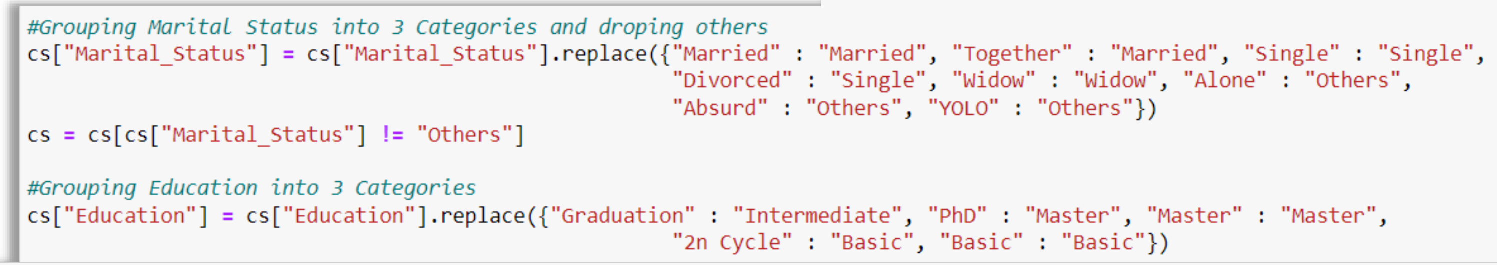
In Age Plot the Response Decreases with Age whereas Complain Increases

Here with High Income Responses Increases and Complain Decreases

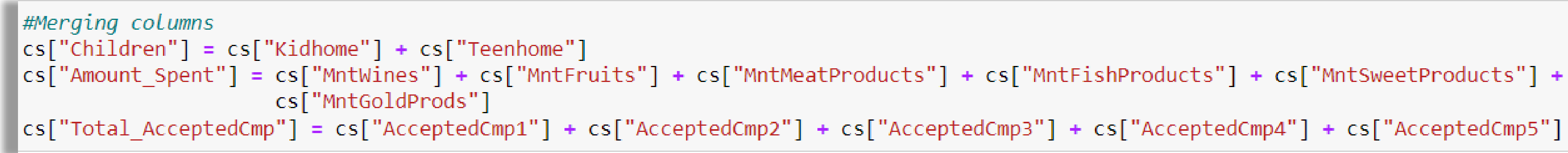
- ○ Created column Age from Birth Year
- ○ Classified Marital Status & Education and then One hot Encoded
- ○ Removed Outliers by dropping them
- ○ Merged columns like Children, Amount Spend & Total_AcceptedCmp
- ○ Dropped irrelevant columns



```
#Calculate Age using Year_Birth Column
CY = pd.to_datetime("today").year
cs["Age"] = CY - cs["Year_Birth"]
```



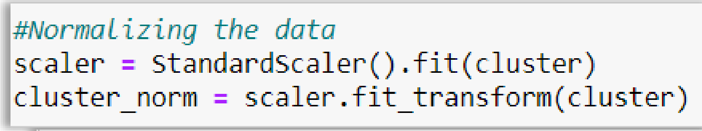
```
#Grouping Marital Status into 3 Categories and droping others
cs["Marital_Status"] = cs["Marital_Status"].replace({"Married" : "Married", "Together" : "Married", "Single" : "Single",
                                                      "Divorced" : "Single", "Widow" : "Widow", "Alone" : "Others",
                                                      "Absurd" : "Others", "YOLO" : "Others"})
cs = cs[cs["Marital_Status"] != "Others"]

#Grouping Education into 3 Categories
cs["Education"] = cs["Education"].replace({"Graduation" : "Intermediate", "PhD" : "Master", "Master" : "Master",
                                            "2n Cycle" : "Basic", "Basic" : "Basic"})
```



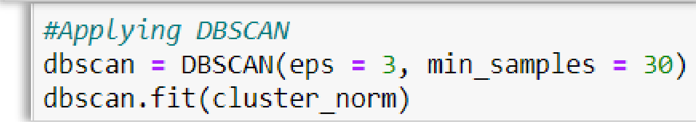
```
#Merging columns
cs["Children"] = cs["Kidhome"] + cs["Teenhome"]
cs["Amount_Spent"] = cs["MntWines"] + cs["MntFruits"] + cs["MntMeatProducts"] + cs["MntFishProducts"] + cs["MntSweetProducts"] +
                     cs["MntGoldProds"]
cs["Total_AcceptedCmp"] = cs["AcceptedCmp1"] + cs["AcceptedCmp2"] + cs["AcceptedCmp3"] + cs["AcceptedCmp4"] + cs["AcceptedCmp5"]
```

Clustering

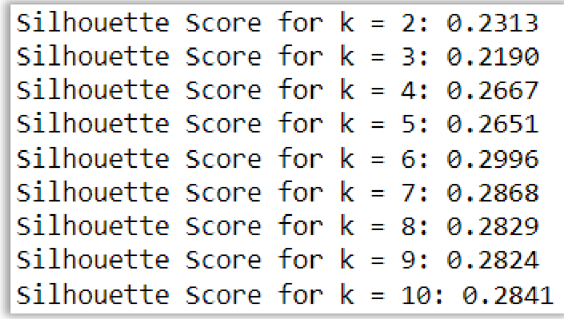
- Standardize the data
- Applied silhouette score on k means & WCSS (Scree Plot) to find the optimal k
- Dropped outliers using DBSCAN



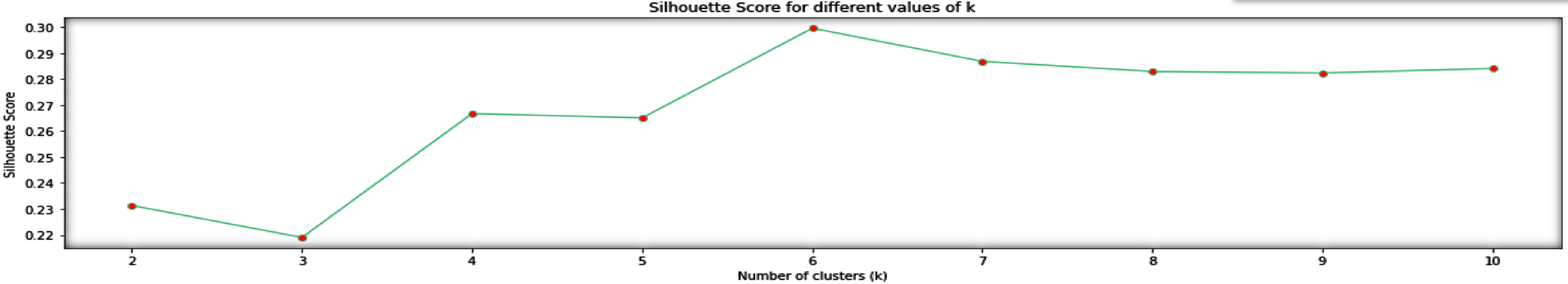
```
#Normalizing the data
scaler = StandardScaler().fit(cluster)
cluster_norm = scaler.fit_transform(cluster)
```



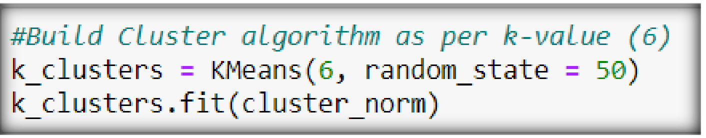
```
#Applying DBSCAN
dbscan = DBSCAN(eps = 3, min_samples = 30)
dbscan.fit(cluster_norm)
```



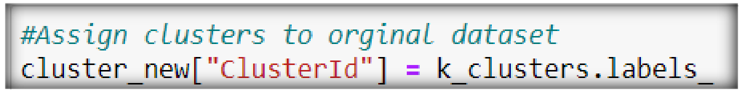
```
Silhouette Score for k = 2: 0.2313
Silhouette Score for k = 3: 0.2190
Silhouette Score for k = 4: 0.2667
Silhouette Score for k = 5: 0.2651
Silhouette Score for k = 6: 0.2996
Silhouette Score for k = 7: 0.2868
Silhouette Score for k = 8: 0.2829
Silhouette Score for k = 9: 0.2824
Silhouette Score for k = 10: 0.2841
```



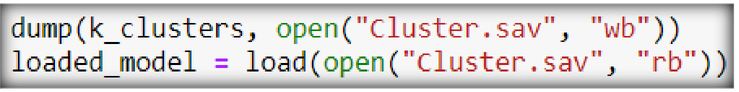
Model Building



```
#Build Cluster algorithm as per k-value (6)
k_clusters = KMeans(6, random_state = 50)
k_clusters.fit(cluster_norm)
```

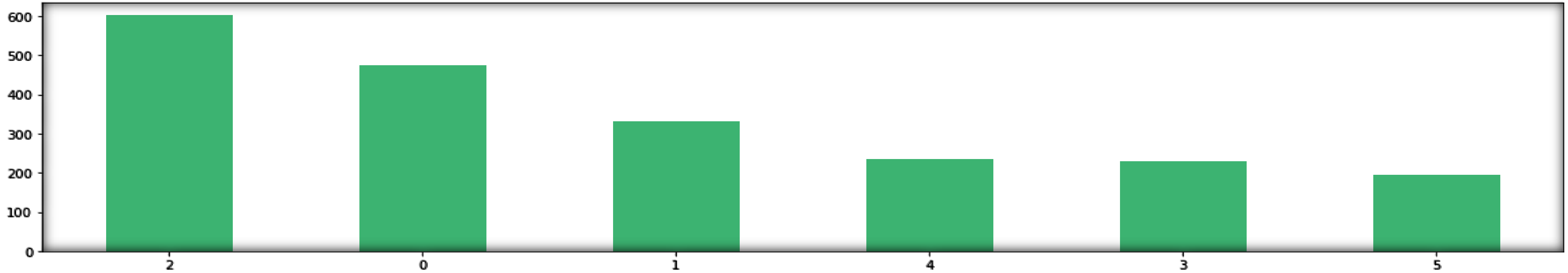


```
#Assign clusters to orginal dataset
cluster_new["ClusterId"] = k_clusters.labels_
```



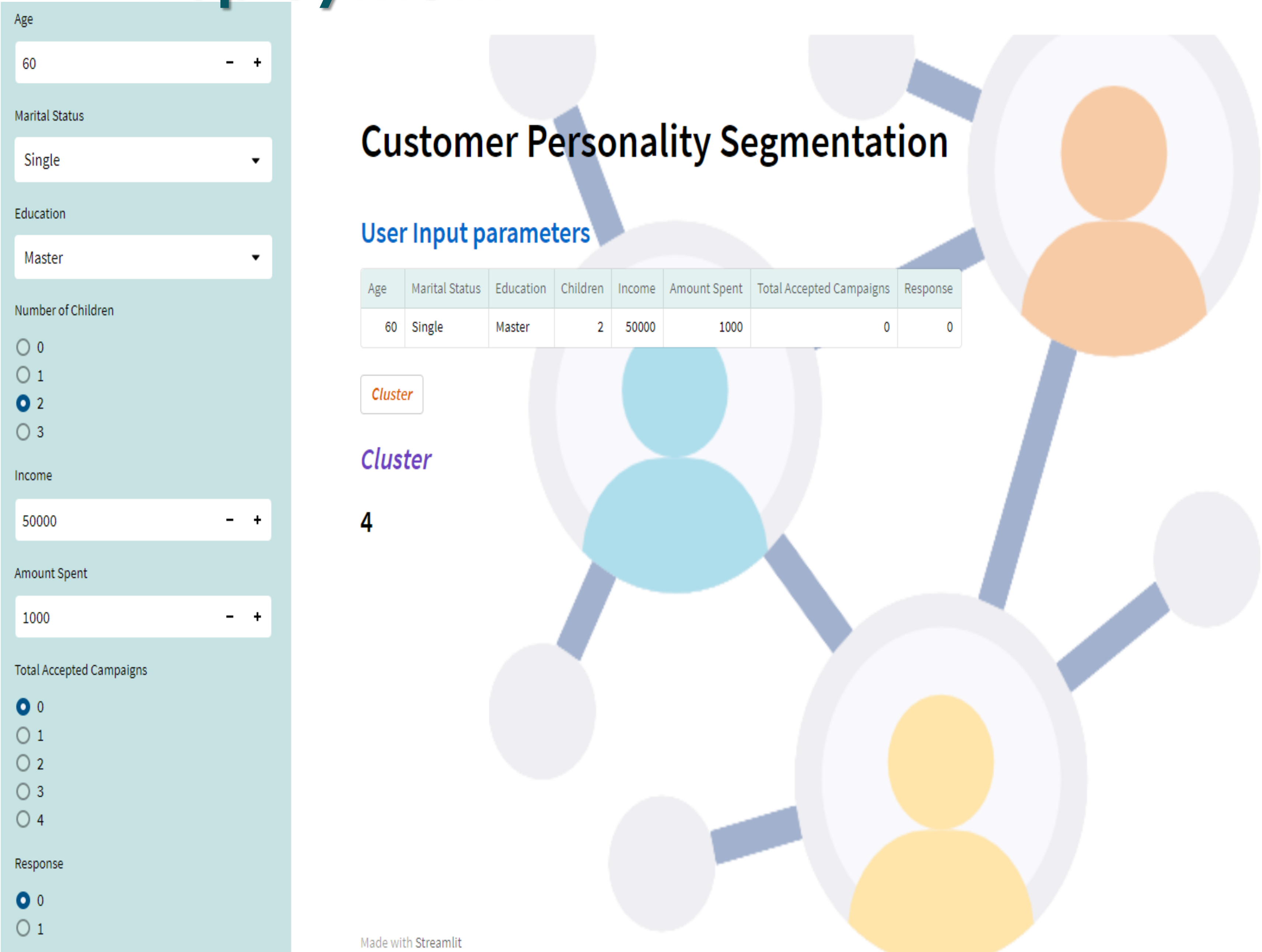
```
dump(k_clusters, open("Cluster.sav", "wb"))
loaded_model = load(open("Cluster.sav", "rb"))
```

- Finalised the model with k = 6 using Kmeans
- Cluster Labels Added to the original Data
- Dumped the model using pickle



Deployment

- Deployed model on *Streamlit* using command prompt
- Age, Marital Status, Education, Children, Income, Amount Spent, Total_AcceptedCmp & Response as user input
- Cluster Button to predict ClusterId



Age

60 - +

Marital Status

Single ▾

Education

Master ▾

Number of Children

○ 0
○ 1
● 2
○ 3

Income

50000 - +

Amount Spent

1000 - +

Total Accepted Campaigns

● 0
○ 1
○ 2
○ 3
○ 4

Response

● 0
○ 1

Customer Personality Segmentation

User Input parameters

Age	Marital Status	Education	Children	Income	Amount Spent	Total Accepted Campaigns	Response
60	Single	Master	2	50000	1000	0	0

Cluster

4

Made with Streamlit

Thank You