

# McDonalds\_casestudy

June 29, 2024

```
[ ]: Mcdonald Market Segmentation Case Study- Kalyani bhosale
```

```
[ ]: import numpy as np
```

```
[53]: import pandas as pd
```

```
[60]: import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.cluster import AgglomerativeClustering
import os
```

```
[62]: from os.path import exists
```

```
[138]: exists('mcdonalds_data.csv')
True
```

```
[138]: True
```

```
[140]: data = pd.read_csv('mcdonalds_data.csv')
```

```
[142]: data.head()
```

```
[142]:
```

	CustomerID	yummy	convenient	spicy	fattening	greasy	fast	cheap	tasty	\
0	1	No	Yes	No	Yes	No	Yes	Yes	No	
1	2	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
2	3	No	Yes	Yes	Yes	Yes	Yes	No	Yes	
3	4	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
4	5	No	Yes	No	Yes	Yes	Yes	Yes	No	

	expensive	healthy	disgusting	Like	Age	VisitFrequency	Gender
0	Yes	No	No	-3	61	Every three months	Female
1	Yes	No	No	2	51	Every three months	Female
2	Yes	Yes	No	1	62	Every three months	Female
3	No	No	Yes	4	69	Once a week	Female
4	No	Yes	No	2	49	Once a month	Male

```
[144]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1453 entries, 0 to 1452
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   CustomerID            1453 non-null   int64
 1   yummy                 1453 non-null   object
 2   convenient            1453 non-null   object
 3   spicy                 1453 non-null   object
 4   fattening             1453 non-null   object
 5   greasy                1453 non-null   object
 6   fast                  1453 non-null   object
 7   cheap                 1453 non-null   object
 8   tasty                 1453 non-null   object
 9   expensive             1453 non-null   object
10   healthy               1453 non-null   object
11   disgusting            1453 non-null   object
12   Like                  1453 non-null   object
13   Age                   1453 non-null   int64
14   VisitFrequency        1453 non-null   object
15   Gender                 1453 non-null   object
dtypes: int64(2), object(14)
memory usage: 181.8+ KB
```

```
[146]: print(pd.isnull(data).sum())
```

```
CustomerID      0
yummy           0
convenient      0
spicy           0
fattening       0
greasy          0
fast            0
cheap           0
tasty           0
expensive       0
healthy         0
disgusting      0
Like            0
Age             0
VisitFrequency  0
Gender          0
dtype: int64
```

```
[148]: data.describe()
```

```
[148]:
```

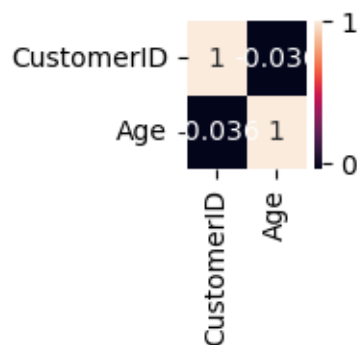
	CustomerID	Age
count	1453.000000	1453.000000
mean	727.000000	44.604955
std	419.589283	14.221178
min	1.000000	18.000000
25%	364.000000	33.000000
50%	727.000000	45.000000
75%	1090.000000	57.000000
max	1453.000000	71.000000

```
[150]: data.corr(numeric_only = True)
```

```
[150]:
```

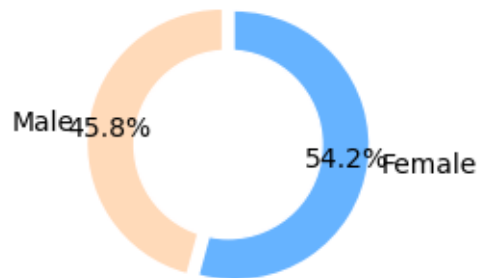
	CustomerID	Age
CustomerID	1.000000	-0.036447
Age	-0.036447	1.000000

```
[124]: plt.figure(figsize=(1,1))
sns.heatmap(data.corr(numeric_only = True), annot=True)
plt.show()
```

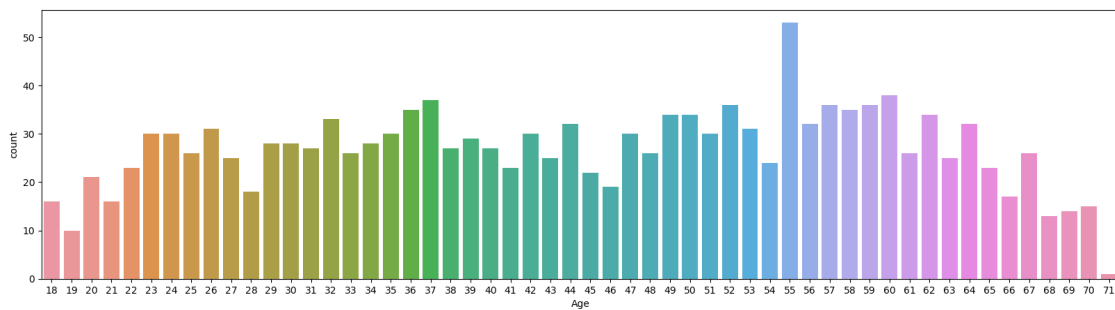


```
[152]: labels = ['Male', 'Female']
sizes = [data.query('Gender == "Male"').Gender.count(), data.query('Gender == "Female"').Gender.count()]
#colors
colors = ['#ffdaB9', '#66b3ff']
#explosion
explode = (0.05, 0.05)
plt.figure(figsize=(2,2))
my_circle=plt.Circle( (0,0), 0.7, color='white')
plt.pie(sizes, colors = colors, labels=labels, autopct='%1.1f%%',
        startangle=90, pctdistance=0.85, explode=explode)
p=plt.gcf()
plt.axis('equal')
```

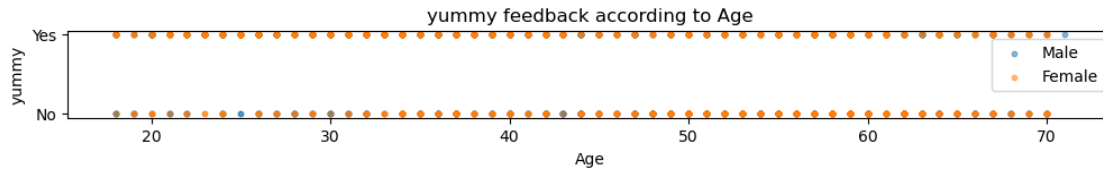
```
p.gca().add_artist(my_circle)
plt.show()
```



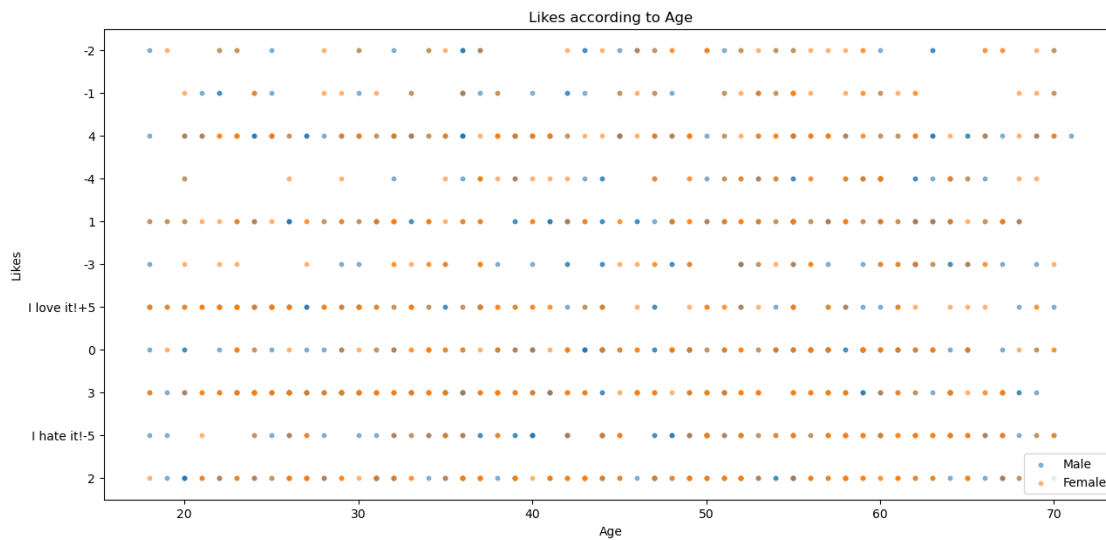
```
[192]: plt.figure(figsize=(20,5))
sns.countplot(x=data.Age)
plt.xlabel("Age")
plt.ylabel("count")
plt.show()
```



```
[214]: plt.figure(figsize=(12,1))
gender = ['Male', 'Female']
for i in gender:
    plt.scatter(x='Age',y='yummy', data=data[data['Gender']==i],s = 10 , alpha=
    ↪ 0.5 , label = i)
plt.legend()
plt.xlabel("Age")
plt.ylabel("yummy")
plt.title("yummy feedback according to Age")
plt.show()
```



```
[216]: plt.figure(figsize=(15,7))
gender = ['Male', 'Female']
for i in gender:
    plt.scatter(x='Age',y='Like', data=data[data['Gender']==i],s = 10 , alpha = 0.5 , label = i)
plt.legend()
plt.xlabel("Age")
plt.ylabel("Likes")
plt.title("Likes according to Age")
plt.show()
```



Implement Clustering Algorithms

K-Means Clustering Algorithm

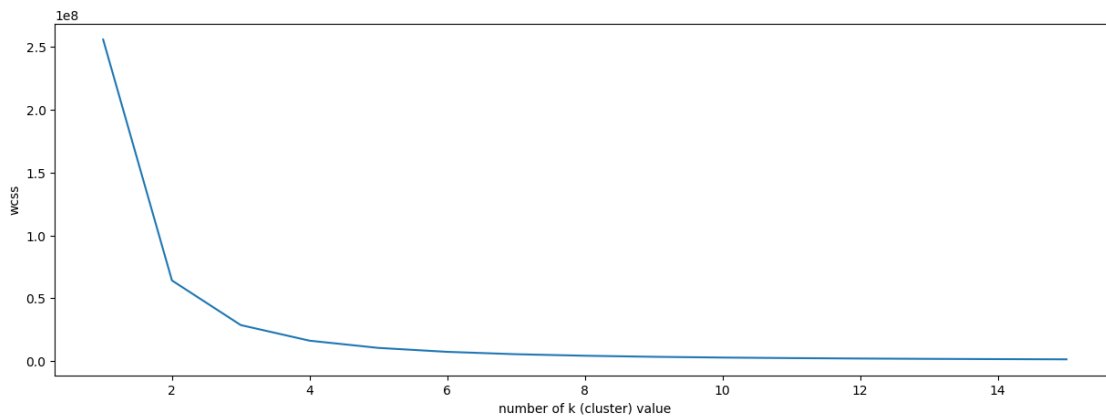
```
[298]: import warnings
warnings.filterwarnings("ignore")
wcss = []
data_model = data.
    drop(['Gender','yummy','convenient','spicy','fattening','greasy','fast','cheap','tasty','expensive'])
for k in range(1,16):
    kmeans = KMeans(n_clusters=k)
```

```

kmeans.fit(data_model)
wcss.append(kmeans.inertia_)

plt.figure(figsize=(15,5))
plt.plot(range(1,16),wcss)
plt.xlabel("number of k (cluster) value")
plt.ylabel("wcss")
plt.show()

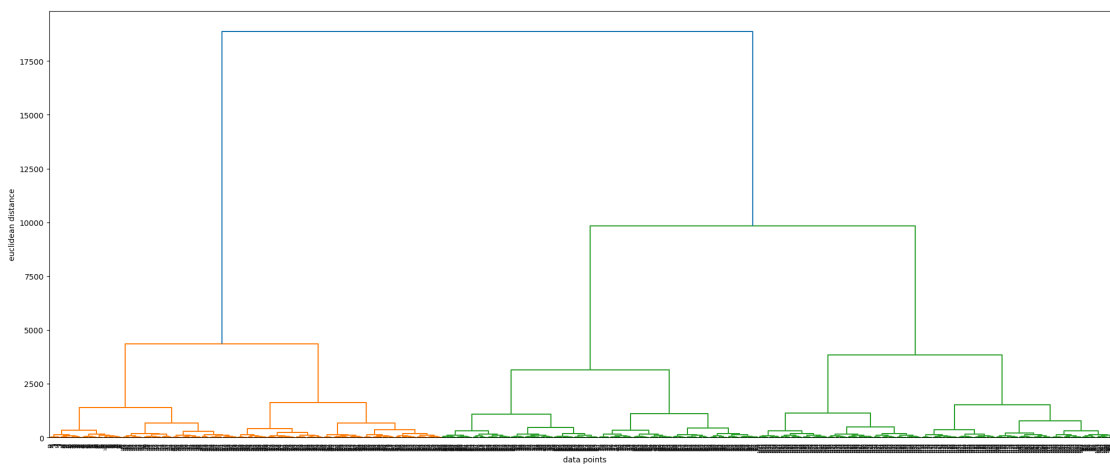
```



```

[315]: merg = linkage(data_model,method="ward")
plt.figure(figsize=(25,10))
dendrogram(merg,leaf_rotation = 90)
plt.xlabel("data points")
plt.ylabel("euclidean distance")
plt.show()

```



[ ]: