```
import pandas  as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import sklearn.metrics as sm


df=pd.read_csv(r"/content/sample_data/FashionSet.csv")


df.describe()
```

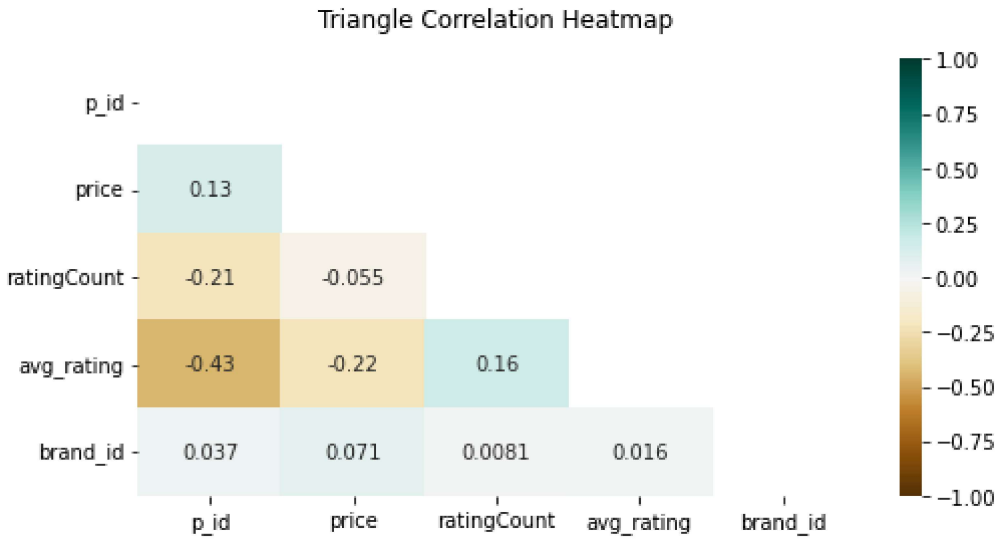
	Unnamed: 0	p_id	price	ratingCount	avg_rating	brand_id
count	14310.000000	1.431000e+04	14310.000000	14310.000000	14310.000000	14310.000000
mean	7154.500000	1.569106e+07	2963.668344	104.509962	2.546139	531.337596
std	4131.085511	3.153509e+06	2564.344664	541.897533	1.661950	292.690867
min	0.000000	7.016600e+04	0.000000	0.000000	0.000000	0.000000
25%	3577.250000	1.413618e+07	1599.000000	2.852941	0.977478	278.000000
50%	7154.500000	1.638216e+07	2199.500000	11.000000	2.823823	551.000000
75%	10731.750000	1.808344e+07	3495.000000	42.000000	4.140389	783.000000
max	14309.000000	1.941576e+07	47999.000000	21274.000000	5.000000	1020.000000

```
df=df.drop(df.columns[[0,9,10]], axis=1)


df.describe()
```

	p_id	price	ratingCount	avg_rating	brand_id
count	1.431000e+04	14310.000000	14310.000000	14310.000000	14310.000000
mean	1.569106e+07	2963.668344	104.509962	2.546139	531.337596
std	3.153509e+06	2564.344664	541.897533	1.661950	292.690867
min	7.016600e+04	0.000000	0.000000	0.000000	0.000000
25%	1.413618e+07	1599.000000	2.852941	0.977478	278.000000
50%	1.638216e+07	2199.500000	11.000000	2.823823	551.000000
75%	1.808344e+07	3495.000000	42.000000	4.140389	783.000000
max	1.941576e+07	47999.000000	21274.000000	5.000000	1020.000000

```
plt.figure(figsize=(8, 4))
# define the mask to set the values in the upper triangle to True
mask = np.triu(np.ones_like(df.corr(), dtype=np.bool))
heatmap = sns.heatmap(df.corr(), mask=mask, vmin=-1, vmax=1, annot=True, cmap='BrBG')
heatmap.set_title('Triangle Correlation Heatmap', fontdict={'fontsize':12}, pad=16);
```

```
<ipython-input-7-d133aed62ca3>:3: DeprecationWarning: `np.bool` is a deprecated alias f
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/rele
  mask = np.triu(np.ones_like(df.corr(), dtype=np.bool))
```




```
df.shape
```

```
(14310, 8)
```

```
a=df.columns.tolist()
print(a)
```

```
['brand_name', 'p_id', 'Product_name', 'price', 'color', 'ratingCount', 'avg_rating', '
```

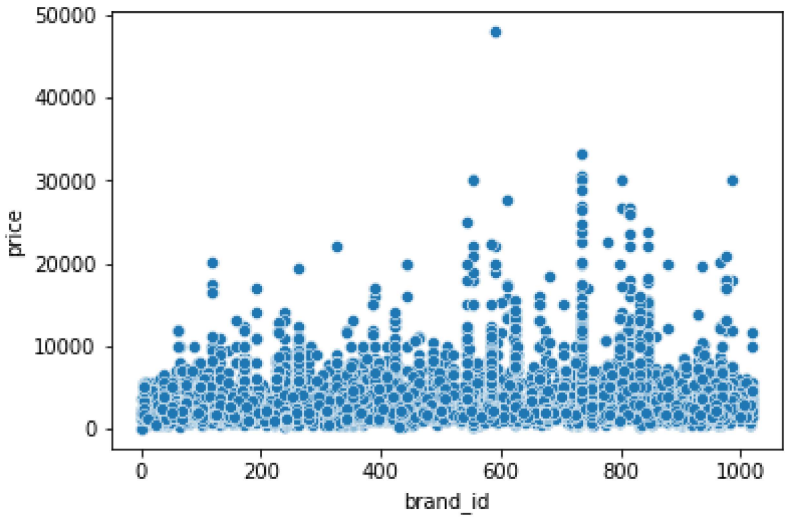
```
df1 = df.drop(df.columns[[0,1,3,2,4,5,6,7]], axis=1)
arn=df.columns.tolist()
print(arn)
```

```
['brand_name', 'p_id', 'Product_name', 'price', 'color', 'ratingCount', 'avg_rating', '
```

```
b=['price']
```

```
for i in b:
```

```
    sns.scatterplot(x="brand_id", y=i ,palette='spectral',data=df, )
    plt.figure(figsize=(15,5))
    plt.show()
    plt.savefig('fig1.png')
```




```
<Figure size 1080x360 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

```
df.describe()
```

	p_id	price	ratingCount	avg_rating	brand_id	mrc
count	1.431000e+04	14310.000000	14310.000000	14310.000000	14310.000000	14310.000000
mean	1.569106e+07	2963.668344	104.509962	2.546139	531.337596	84.830455
std	3.153509e+06	2564.344664	541.897533	1.661950	292.690867	181.167633
min	7.016600e+04	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.413618e+07	1599.000000	2.852941	0.977478	278.000000	2.500000
50%	1.638216e+07	2199.500000	11.000000	2.823823	551.000000	14.720721
75%	1.808344e+07	3495.000000	42.000000	4.140389	783.000000	99.107143
max	1.941576e+07	47999.000000	21274.000000	5.000000	1020.000000	6076.000000

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt


scaler=MinMaxScaler()
scaler.fit(df[['price']])
df[['price']]= scaler.transform(df[['price']])
df
```

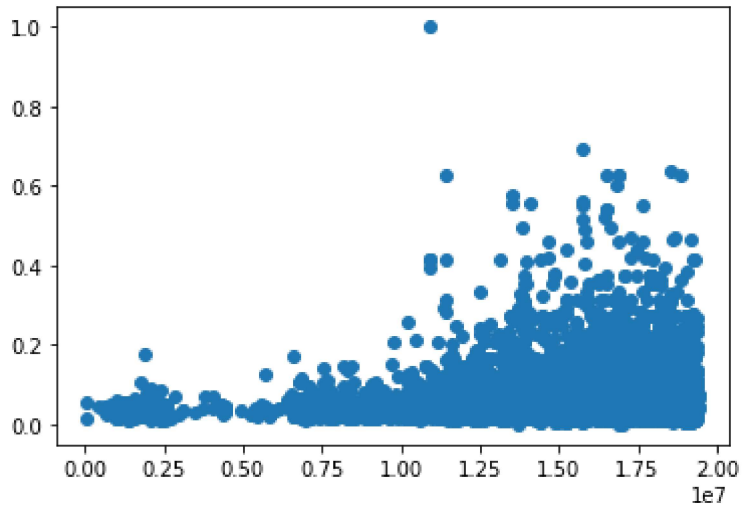
	brand_name	p_id	Product_name	price	color	ratingCount	avg_rating
0	DUPATTA BAZAAR	1518329	Dupatta Bazaar White Embroidered Chiffon Dupatta	0.018730	White	1321.000000	4.548827
1	ROADSTER	5829334	Roadster Women Mustard Yellow Solid Hooded Swe...	0.024980	Mustard	5462.000000	4.313255
2	INDDUS	10340119	Inddus Peach-Coloured & Beige Unstitched Dress...	0.120815	Peach	145.000000	4.068966
3	SASSAFRAS	10856380	SASSAFRAS Women Black Parallel Trousers	0.031230	Black	9124.000000	4.147523
4	KOTTY	12384822	Kotty Women Black Wide Leg High-Rise Clean Loo...	0.041647	Black	12260.000000	4.078467
...	...	...	...	...	...	...	...
14305	THE CHENNAI	17029604	The Chennai Silks Pink &	0.083314	Pink	0.000000	0.000000

```
scaler=MinMaxScaler()
scaler.fit(df[['ratingCount']])
df[['ratingCount']]= scaler.transform(df[['ratingCount']])
df
```

	brand_name	p_id	Product_name	price	color	ratingCount	avg_rating
0	DUPATTA BAZAAR	1518329	Dupatta Bazaar White Embroidered Chiffon Dupatta	0.018730	White	0.062095	4.548827
1	ROADSTER	5829334	Roadster Women Mustard Yellow Solid Hooded Swe...	0.024980	Mustard	0.256745	4.313255
			Inddus Peach-Coloured &				

```
plt.scatter(df['p_id'],df['price'])
```

```
<matplotlib.collections.PathCollection at 0x7f5b82bb0d90>
```




```
km= KMeans(n_clusters=4, random_state=0)
```

```
df1= df[['price','brand_id']]
df1
```

	price	brand_id
0	899.0	242.0
1	1199.0	750.0
2	5799.0	389.0
3	1499.0	783.0
4	1999.0	482.0

```
predict_y= km.fit_predict(df1)
predict_y
#km.labels_
```

```
array([0, 0, 3, ..., 0, 0, 0], dtype=int32)
```

```
km.cluster_centers_
```

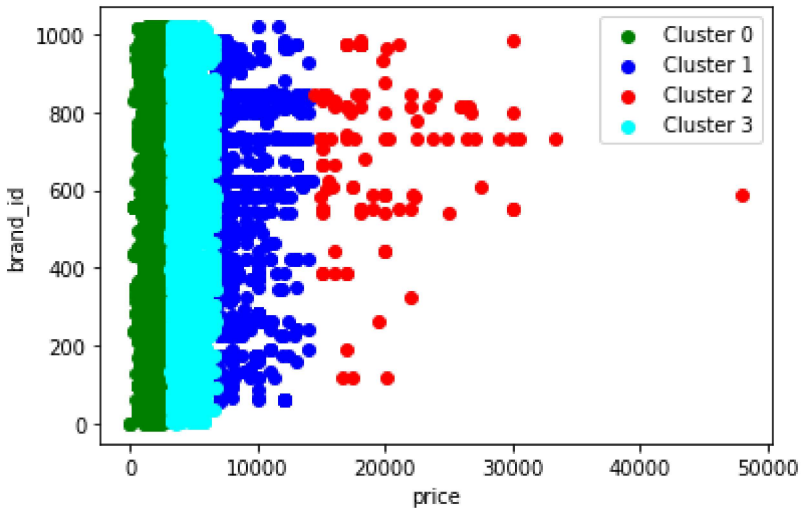
```
array([[ 1863.03815828,   522.59349752],
       [ 8980.81303813,   569.78843788],
       [19760.5826087 ,   708.53913043],
       [ 4405.67224188,   543.62110003]])
```

```
df['clusterBrandnPrice']=predict_y
df
```

	brand_name	p_id	Product_name	price	color	ratingCount	avg_rating	b
0	DUPATTA BAZAAR	1518329	Dupatta Bazaar White Embroidered Chiffon Dupatta	899.0	White	1321.000000	4.548827	

```
df_c0=df[df.clusterBrandnPrice==0]
df_c1=df[df.clusterBrandnPrice==1]
df_c2=df[df.clusterBrandnPrice==2]
df_c3=df[df.clusterBrandnPrice==3]
plt.scatter(df_c0['price'], df_c0['brand_id'], label='Cluster 0', color='green')
plt.scatter(df_c1['price'], df_c1['brand_id'], label='Cluster 1',color='blue')
plt.scatter(df_c2['price'], df_c2['brand_id'], label='Cluster 2',color='red')
plt.scatter(df_c3['price'], df_c3['brand_id'], label='Cluster 3',color='aqua')

plt.xlabel('price')
plt.ylabel('brand_id')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fec6080a820>
```



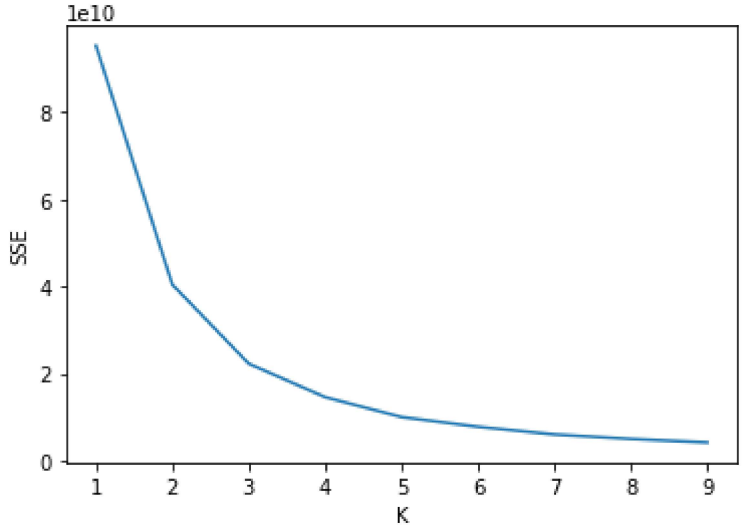

```
#Elbow method
#Calculate Sum of Square Error(SSE)

k_range=range(1,10)
sse=[]
for k in k_range:
    km = KMeans(n_clusters=k)
    km.fit(df[['price', 'brand_id']])
    sse.append(km.inertia_)

sse
```

```
[95319854206.03316,
 40449583071.90943,
```

```
22277827648.218903,
14605202141.05001,
10023622837.988148,
7793518172.808864,
6026081467.50687,
4994524848.944635,
4186024954.613204]
```

```
plt.xlabel('K')
plt.ylabel('SSE')
plt.plot(k_range,sse)
```

```
[<matplotlib.lines.Line2D at 0x7fec6063ad60>]
```




```
 a=df.columns.tolist()
print(a)
```

```
['brand_name', 'p_id', 'Product_name', 'price', 'color', 'ratingCount', 'avg_rating', '
```

```
df
```

	brand_name	p_id	Product_name	price	color	ratingCount	avg_rating	b
0	DUPATTA BAZAAR	1518329	Dupatta Bazaar White Embroidered Chiffon Dupatta	899.0	White	1321.000000	4.548827	
1	ROADSTER	5829334	Roadster Women Mustard Yellow Solid Hooded Swe...	1199.0	Mustard	5462.000000	4.313255	
2	INDDUS	10340119	Inddus Peach-Coloured & Beige Unstitched Dress...	5799.0	Peach	145.000000	4.068966	
			SASSAFRAS					

```
df.to_csv(r'hivefile.csv')
```

			Trousers					

hierarchical clustering

4	KOTTY	12384822	. [illegible]	1999.0	Black	12260.000000	4.078467	

```
from sklearn.datasets import make_blobs
```

```
df.head(5)
```

	brand_name	p_id	Product_name	price	color	ratingCount	avg_rating	brand_
0	DUPATTA BAZAAR	1518329	Dupatta Bazaar White Embroidered Chiffon Dupatta	899.0	White	1321.0	4.548827	24
1	ROADSTER	5829334	Roadster Women Mustard Yellow Solid Hooded Swe...	1199.0	Mustard	5462.0	4.313255	75

```
df2=df.drop(df.columns[[0,2,4]], axis=1)
df2
```

	p_id	price	ratingCount	avg_rating	brand_id	clusterBrandnPrice
0	1518329	899.0	1321.000000	4.548827	242.0	0
1	5829334	1199.0	5462.000000	4.313255	750.0	0
2	10340119	5799.0	145.000000	4.068966	389.0	3
3	10856380	1499.0	9124.000000	4.147523	783.0	0
4	12384822	1999.0	12260.000000	4.078467	482.0	0
...	...	...	...	...	...	...
14305	17029604	3999.0	0.000000	0.000000	880.0	3
14306	17600212	2050.0	0.000000	0.000000	471.0	0
14307	18159266	1659.0	1.190476	0.653061	475.0	0
14308	18921114	2399.0	51.605263	2.654093	404.0	0

```
from sklearn.preprocessing import normalize
data_scaled = normalize(df2)
data_scaled = pd.DataFrame(data_scaled, columns=df2.columns)
data_scaled.head()
```

	p_id	price	ratingCount	avg_rating	brand_id	clusterBrandnPrice
0	0.999999	0.000592	0.000870	2.995941e-06	0.000159	0.000000e+00
1	1.000000	0.000206	0.000937	7.399221e-07	0.000129	0.000000e+00
2	1.000000	0.000561	0.000014	3.935124e-07	0.000038	2.901320e-07
3	1.000000	0.000138	0.000840	3.820354e-07	0.000072	0.000000e+00
4	0.999999	0.000161	0.000990	3.293115e-07	0.000039	0.000000e+00

Price and RatingCount Hierarchical Clustering

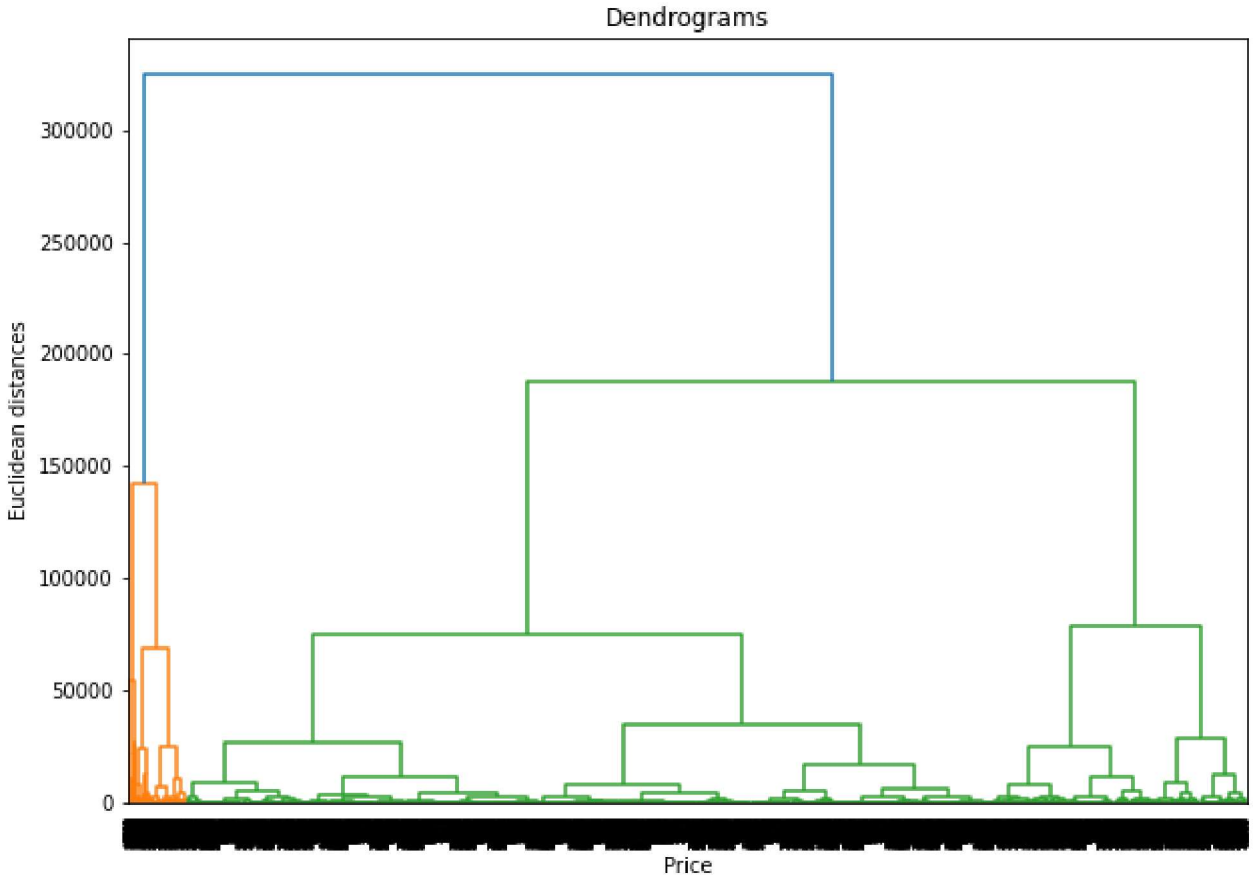
```
x = df2.iloc[:, [1]].values
y=df2.iloc[:,2].values
```

```
print(y)
print(x)
```

```
[1.32100000e+03 5.46200000e+03 1.45000000e+02 ... 1.19047619e+00
 5.16052632e+01 1.03092784e-01]
[[ 899.]
 [1199.]
 [5799.]
 ...
```

```
       [1659.]
       [2399.]
       [2599.]]
```

```
import scipy.cluster.hierarchy as shc
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
plt.xlabel('Price')
plt.ylabel('Euclidean distances')
dend = shc.dendrogram(shc.linkage(x, method='ward'))
```




```
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
```

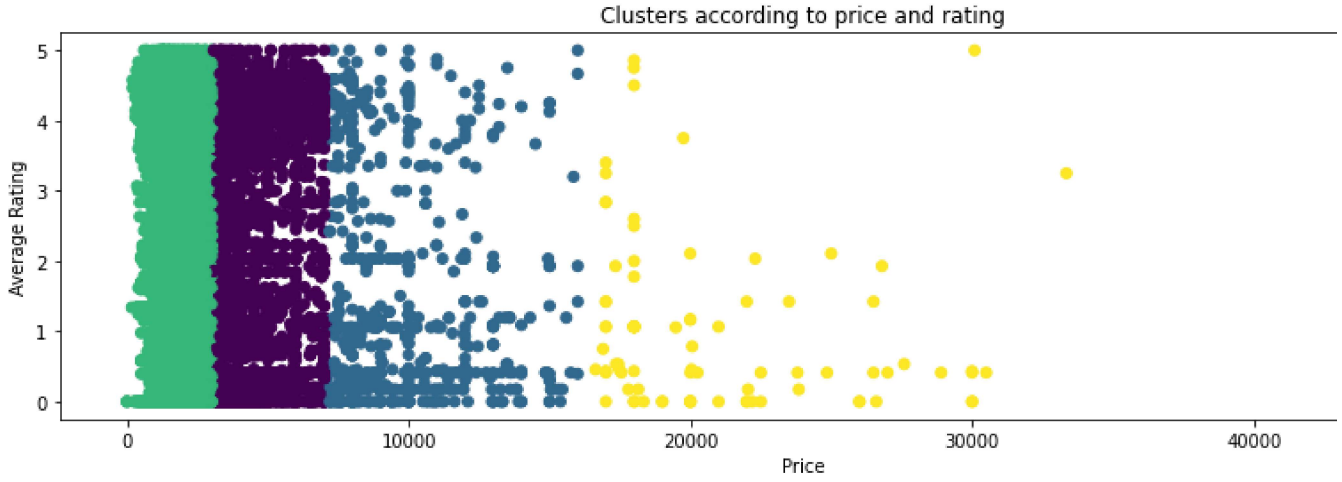
```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=4, affinity='euclidean', linkage='ward')
cluster.fit_predict(x)
```

```
     array([2, 2, 0, ..., 2, 2, 2])
```

```
plt.figure(figsize=(15, 4))
plt.scatter(df['price'], df['avg_rating'], c=cluster.labels_)
plt.title('Clusters according to price and rating')
```

```
plt.xlabel('Price')
plt.ylabel('Average Rating')
```

```
Text(0, 0.5, 'Average Rating')
```




```
plt.figure(figsize=(15, 4))
plt.scatter(df['price'], df['ratingCount'], c=cluster.labels_)
plt.title('Clusters according to price and ratingCount')
plt.xlabel('Price')
plt.ylabel('RatingCount')
```

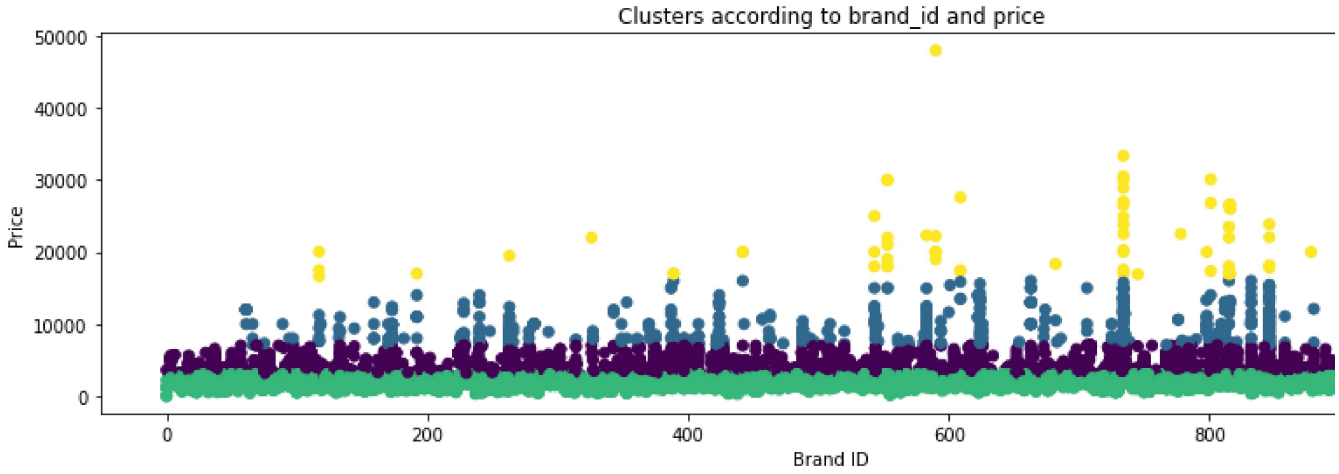
```
Text(0, 0.5, 'RatingCount')
```


```
plt.figure(figsize=(15, 4))
plt.scatter(df['brand_id'], df['avg_rating'], c=cluster.labels_)
plt.title('Clusters according to brand_id and average rating')
plt.xlabel('Brand ID')
plt.ylabel('Average Rating')
```

```
Text(0, 0.5, 'Average Rating')
```


```
plt.figure(figsize=(15, 4))
plt.scatter(df['brand_id'], df['price'], c=cluster.labels_)
plt.title('Clusters according to brand_id and price')
plt.xlabel('Brand ID')
plt.ylabel('Price')
```

```
Text(0, 0.5, 'Price')
```

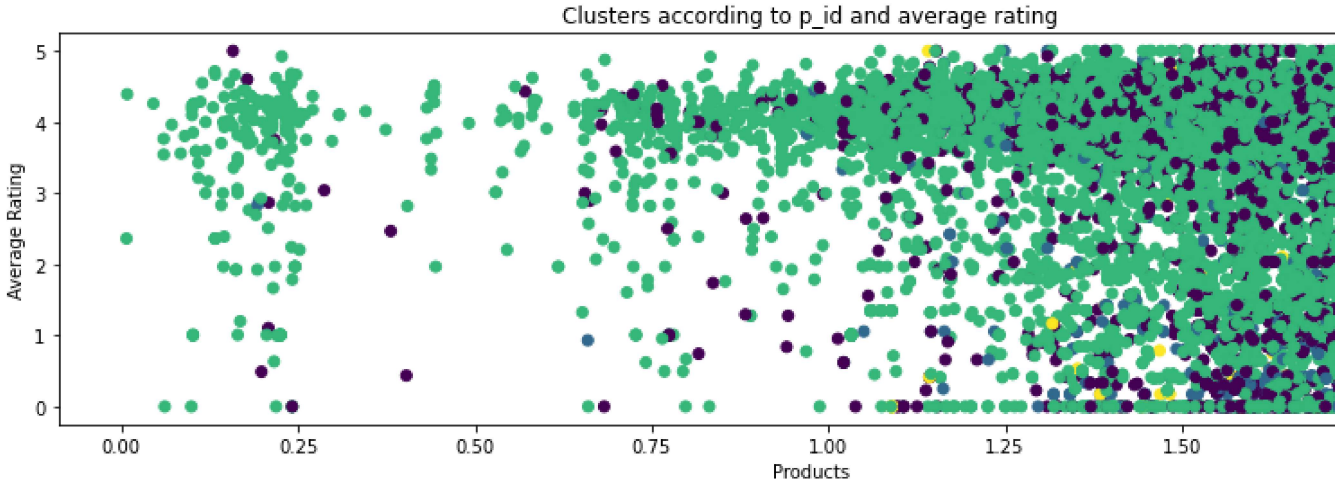



```
plt.figure(figsize=(15, 4))
plt.scatter(df['p_id'], df['price'], c=cluster.labels_)
plt.title('Clusters according to p_id and price')
plt.xlabel('Products')
plt.ylabel('Price')
```

```
Text(0, 0.5, 'Price')
```

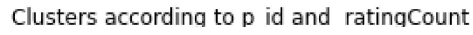

```
plt.figure(figsize=(15, 4))
plt.scatter(df['p_id'], df['avg_rating'], c=cluster.labels_)
plt.title('Clusters according to p_id and average rating')
plt.xlabel('Products')
plt.ylabel('Average Rating')
```

```
Text(0, 0.5, 'Average Rating')
```



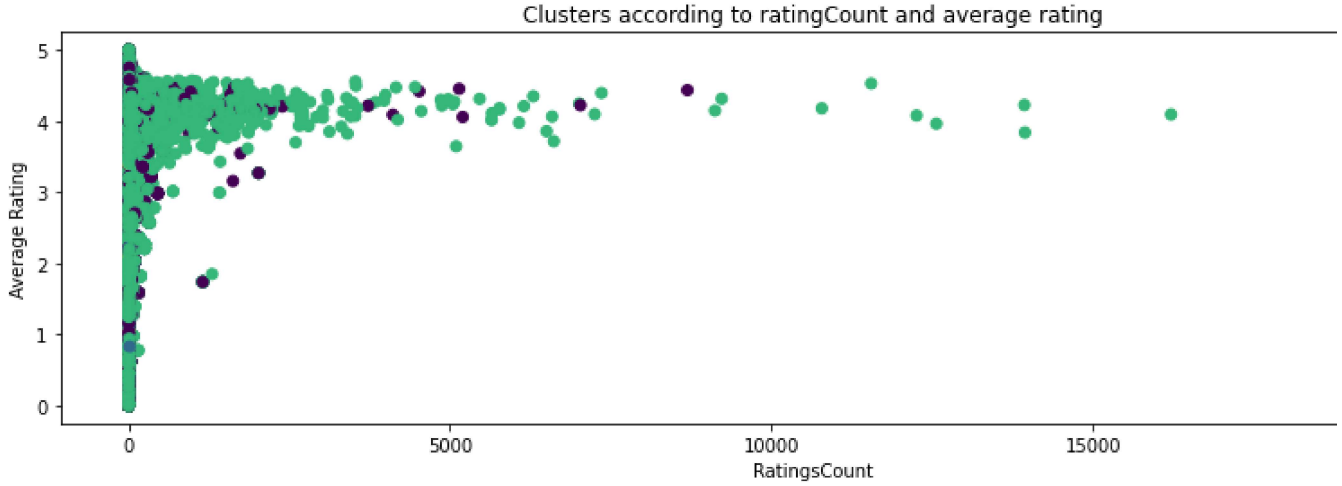

```
plt.figure(figsize=(15, 4))
plt.scatter(df['p_id'], df['ratingCount'], c=cluster.labels_)
plt.title('Clusters according to p_id and  ratingCount')
plt.xlabel('Products')
plt.ylabel('RatingsCount')
```

```
Text(0, 0.5, 'RatingsCount')
```




```
plt.figure(figsize=(15, 4))
plt.scatter(df['ratingCount'], df['avg_rating'], c=cluster.labels_)
plt.title('Clusters according to ratingCount and average rating')
plt.xlabel('RatingsCount')
plt.ylabel('Average Rating')
```

```
Text(0, 0.5, 'Average Rating')
```




```
#to predict which point belongs to which, fit hc objects to data, saves new clusters for char
y_hc=cluster.fit_predict(x)
```

```
print(y_hc)
print(x)
```

```
[2 2 0 ... 2 2 2]
[[ 899.]
 [1199.]
 [5799.]
 ...
 [1659.]
 [2399.]
 [2599.]]
```

```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from math import sqrt

RMSE = float(format(np.sqrt(mean_squared_error(y, y_hc)),'.3f'))
r2 = r2_score(y, y_hc)
print('RMSE =',RMSE, '\nR2 =', r2)
```

```
RMSE = 551.547
R2 = -0.036004047312368614
```

```
!pip install "coclust[alldeps]"
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Collecting coclust[alldeps]
  Downloading coclust-0.2.1.tar.gz (21 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from co
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from co
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (
Requirement already satisfied: matplotlib>=1.5 in /usr/local/lib/python3.8/dist-package
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/l
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packa
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from
Building wheels for collected packages: coclust
  Building wheel for coclust (setup.py) ... done
  Created wheel for coclust: filename=coclust-0.2.1-py3-none-any.whl size=29870 sha256=
  Stored in directory: /root/.cache/pip/wheels/21/e9/ac/5d162f4eb117a4437f05c48020e114b
Successfully built coclust
Installing collected packages: coclust
Successfully installed coclust-0.2.1
```

Colab paid products - Cancel contracts here

✓ 0s completed at 7:12 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.