Freelancing Platform Project:

1. Perform necessary EDA on the data
2. use machine learning to create cluster of similar project

In [1]:
```python
#import necesasry libraries:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

#read dataset:
df = pd.read_csv('Freelance Platform Projects.csv')

#display first five records of dataset:
df.head()
```

Out[1]:

	Title	Category Name	Experience	Sub Category Name	Currency	Budget	Location	Freelancer Preferred From	Type	Date Posted	Description	Duration	Client Registration Date	Client City	Client Country	Client Currency	Client Job Title
0	Banner images for web desgin websites	Design	Entry ($)	Graphic Design	EUR	60.0	remote	ALL	fixed_price	2023-04-29 18:06:39	We are looking to improve the banner images on...	NaN	2010-11-03	Dublin	Ireland	EUR	PPC Management
1	Make my picture a solid silhouette	Video, Photo & Image	Entry ($)	Image Editing	GBP	20.0	remote	ALL	fixed_price	2023-04-29 17:40:28	Hello \n\nI need a quick designer to make 4 pi...	NaN	2017-02-21	London	United Kingdom	GBP	Office manager
2	Bookkeeper needed	Business	Entry ($)	Finance & Accounting	GBP	12.0	remote	ALL	fixed_price	2023-04-29 17:40:06	Hi - I need a bookkeeper to assist with bookke...	NaN	2023-04-09	London	United Kingdom	GBP	Paralegal
3	Accountant needed	Business	Entry ($)	Tax Consulting & Advising	GBP	14.0	remote	ALL	fixed_price	2023-04-29 17:32:01	Hi - I need an accountant to assist me with un...	NaN	2023-04-09	London	United Kingdom	GBP	Paralegal
4	Guest Post on High DA Website	Digital Marketing	Expert ($$$)	SEO	USD	10000.0	remote	ALL	fixed_price	2023-04-29 17:09:36	Hi, I am currently running a project where I w...	NaN	2016-07-01	Mumbai	India	USD	Guest posts buyer

In [2]:
```python
#display rows and columns of adtaset respectively(rows,columns):
df.shape
```

Out[2]:
```
(12222, 17)
```

In [3]:
```python
#display information of data set:
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12222 entries, 0 to 12221
Data columns (total 17 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Title                      12222 non-null  object 
 1   Category Name              12222 non-null  object 
 2   Experience                 12222 non-null  object 
 3   Sub Category Name          12222 non-null  object 
 4   Currency                   12222 non-null  object 
 5   Budget                     12222 non-null  float64
 6   Location                   12222 non-null  object 
 7   Freelancer Preferred From  12222 non-null  object 
 8   Type                       12222 non-null  object 
 9   Date Posted                12222 non-null  object 
 10  Description                12222 non-null  object 
 11  Duration                   1602 non-null   object 
 12  Client Registration Date   12222 non-null  object 
 13  Client City                12222 non-null  object 
 14  Client Country             12222 non-null  object 
 15  Client Currency            12222 non-null  object 
 16  Client Job Title           4588 non-null   object 
dtypes: float64(1), object(16)
memory usage: 1.6+ MB
```

EDA and Preprocessing:

In [4]:
```python
#identifying null values:
df.isna().sum()
```

Out[4]:
```
Title                            0
Category Name                    0
Experience                       0
Sub Category Name                0
Currency                         0
Budget                           0
Location                         0
Freelancer Preferred From        0
Type                             0
Date Posted                      0
Description                      0
Duration                     10620
Client Registration Date         0
Client City                      0
Client Country                   0
Client Currency                  0
Client Job Title              7634
dtype: int64
```

In [5]:
```python
#display datatypes of columns:
df.dtypes
```

Out[5]:
```
Title                         object
Category Name                 object
Experience                    object
Sub Category Name             object
Currency                      object
Budget                       float64
Location                      object
Freelancer Preferred From     object
Type                          object
Date Posted                   object
Description                   object
Duration                      object
Client Registration Date      object
Client City                   object
Client Country                object
Client Currency               object
Client Job Title              object
dtype: object
```

In [6]:
```python
#changing datatype of columns using labelencoder:
encode = ['Title','Category Name','Sub Category Name','Freelancer Preferred From','Description',
                     'Client City','Client Country','Type','Date Posted','Duration','Client Job Title','Currency','Location','Experience','Client Registration Date','Client Currenc

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for column in encode:
    df[column] = le.fit_transform(df[column])
```

In [7]:
```python
#display datatypes of column
df.dtypes
```

Out[7]:
```
Title                          int32
Category Name                  int32
Experience                     int32
Sub Category Name              int32
Currency                       int32
Budget                       float64
Location                       int32
Freelancer Preferred From      int32
Type                           int32
Date Posted                    int32
Description                    int32
Duration                       int32
Client Registration Date       int32
Client City                    int32
Client Country                 int32
Client Currency                int32
Client Job Title               int32
dtype: object
```

In [8]:
```python
#dropping unnecessary columns:
df.drop(columns=['Duration','Client Job Title'],inplace=True)
```

In [9]:
```python
#number of rows and columns in dataset respectively(rows,Coumns)
df.shape
```

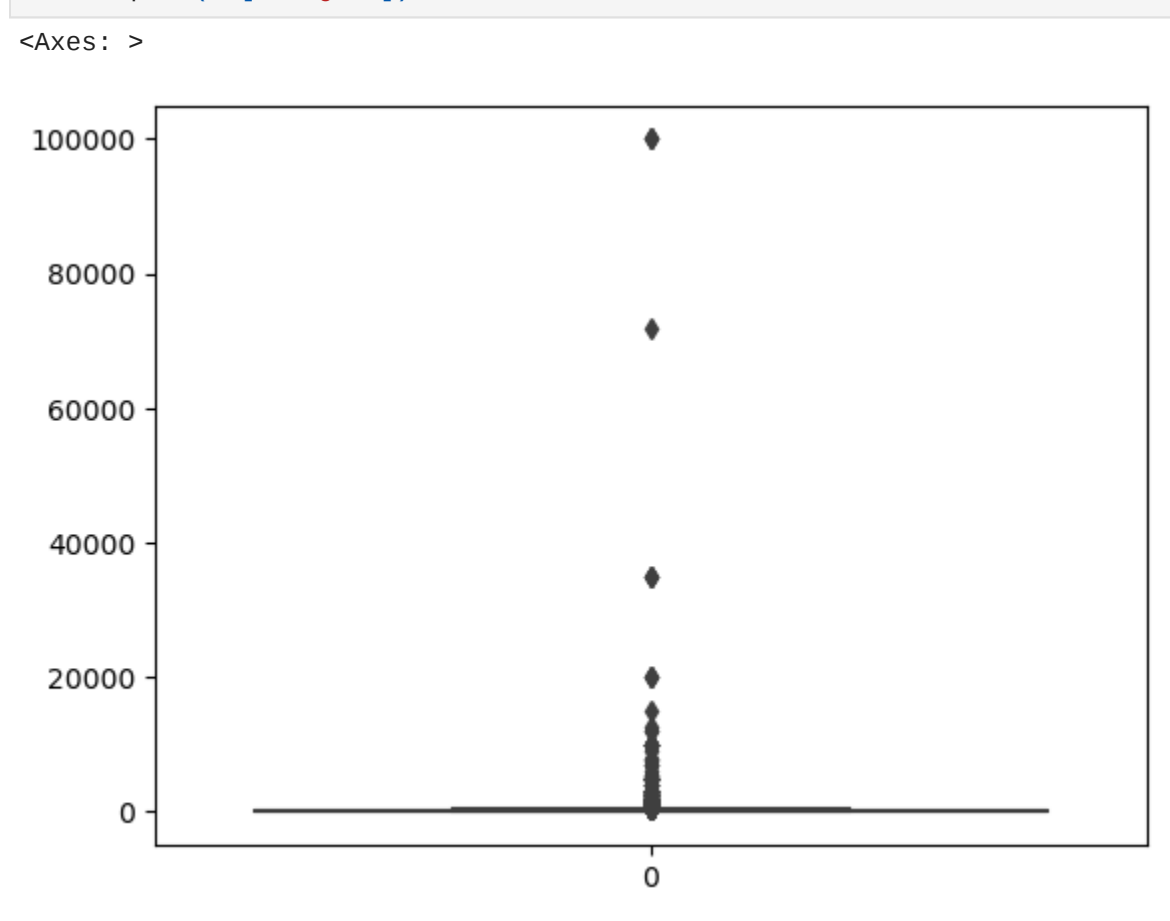
Out[9]:
```
(12222, 15)
```

In [10]:
```python
#display columns of dataset:
df.columns
```

Out[10]:
```
Index(['Title', 'Category Name', 'Experience', 'Sub Category Name', 'Currency',
       'Budget', 'Location', 'Freelancer Preferred From', 'Type',
       'Date Posted', 'Description', 'Client Registration Date', 'Client City',
       'Client Country', 'Client Currency'],
      dtype='object')
```

In [11]:
```python
#identifying outliers and Handling:
import seaborn as sns
import matplotlib.pyplot as plt
sns.boxplot(df['Budget'])
```

Out[11]:
```
<Axes: >
```



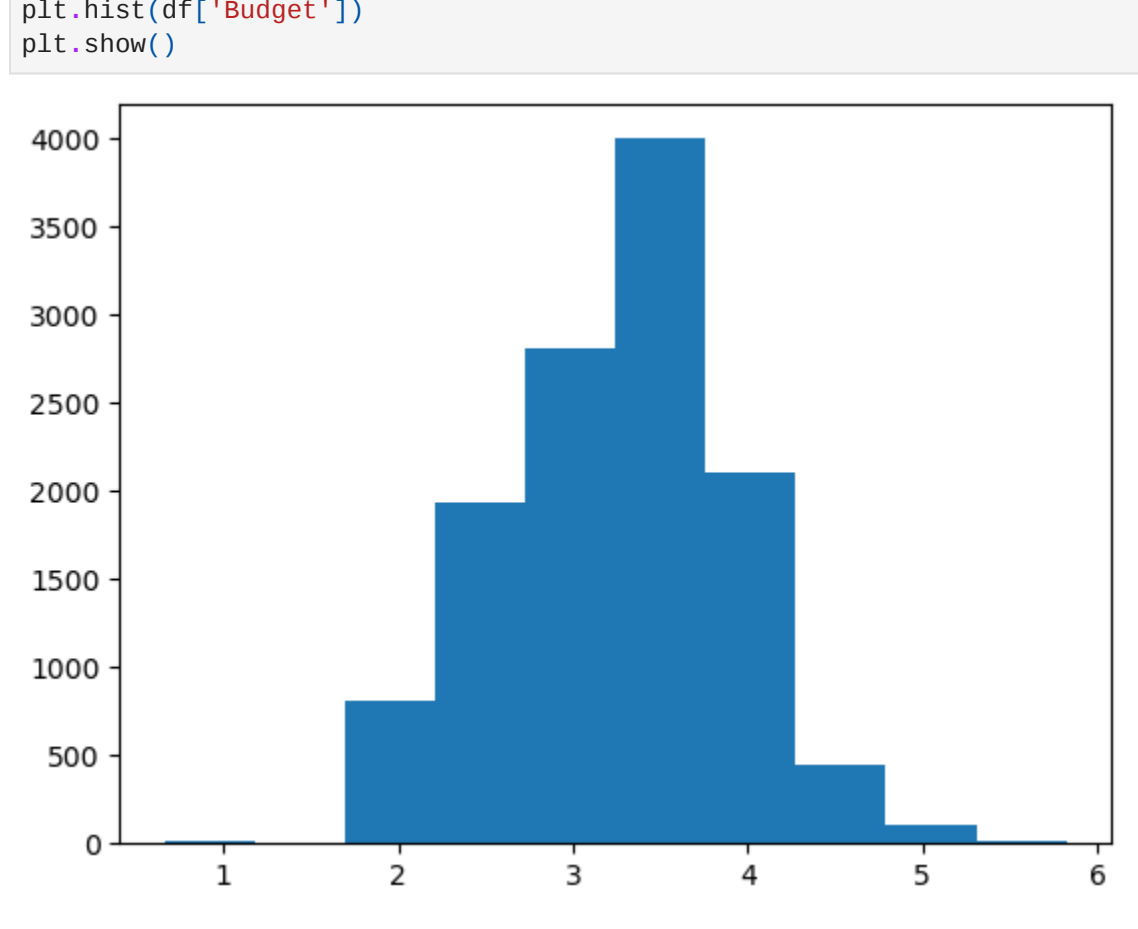
In [12]:
```python
#handling skewness:
df['Budget'].skew()
```

Out[12]:
```
42.455398395555996
```

In [13]:
```python
#handling skewness:
from scipy.stats import boxcox
df['Budget'] = df['Budget']+1
df['Budget'] = boxcox(df['Budget']+1)[0]
df['Budget'].skew()
```

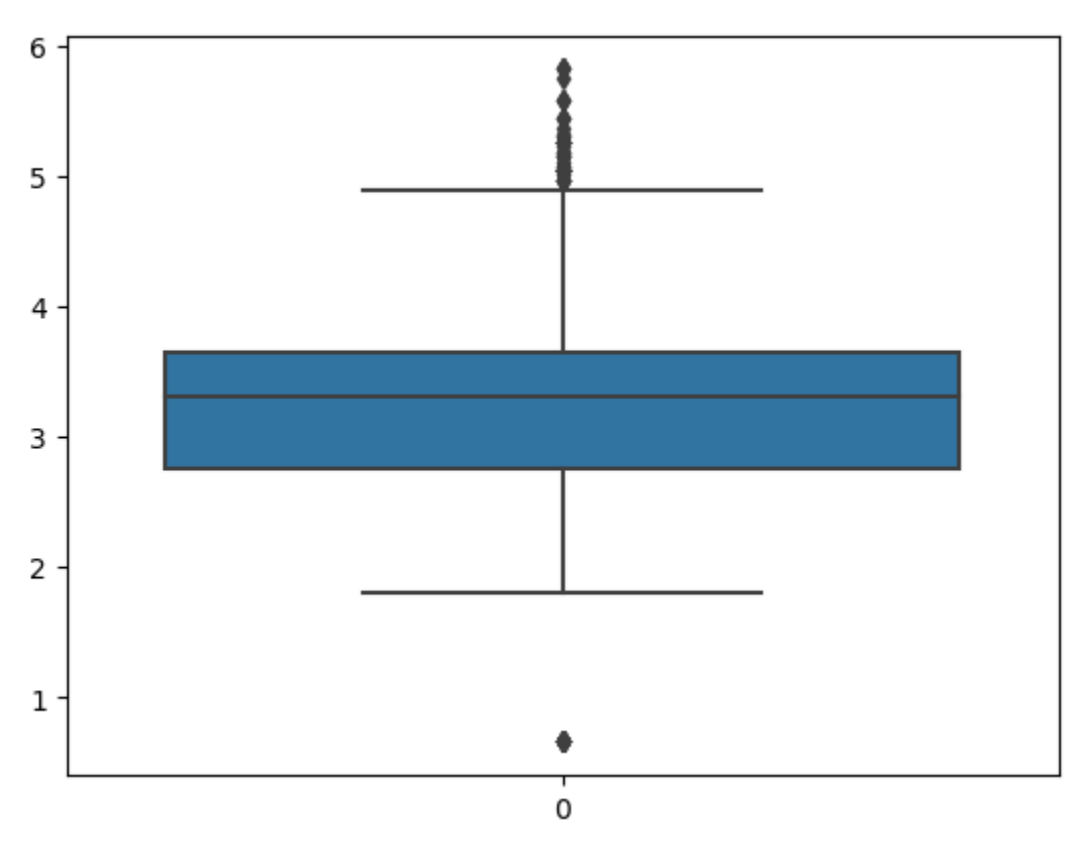
Out[13]:
```
-0.010182795811508298
```

In [14]:
```python
#plotting histogram:
plt.hist(df['Budget'])
plt.show()
```



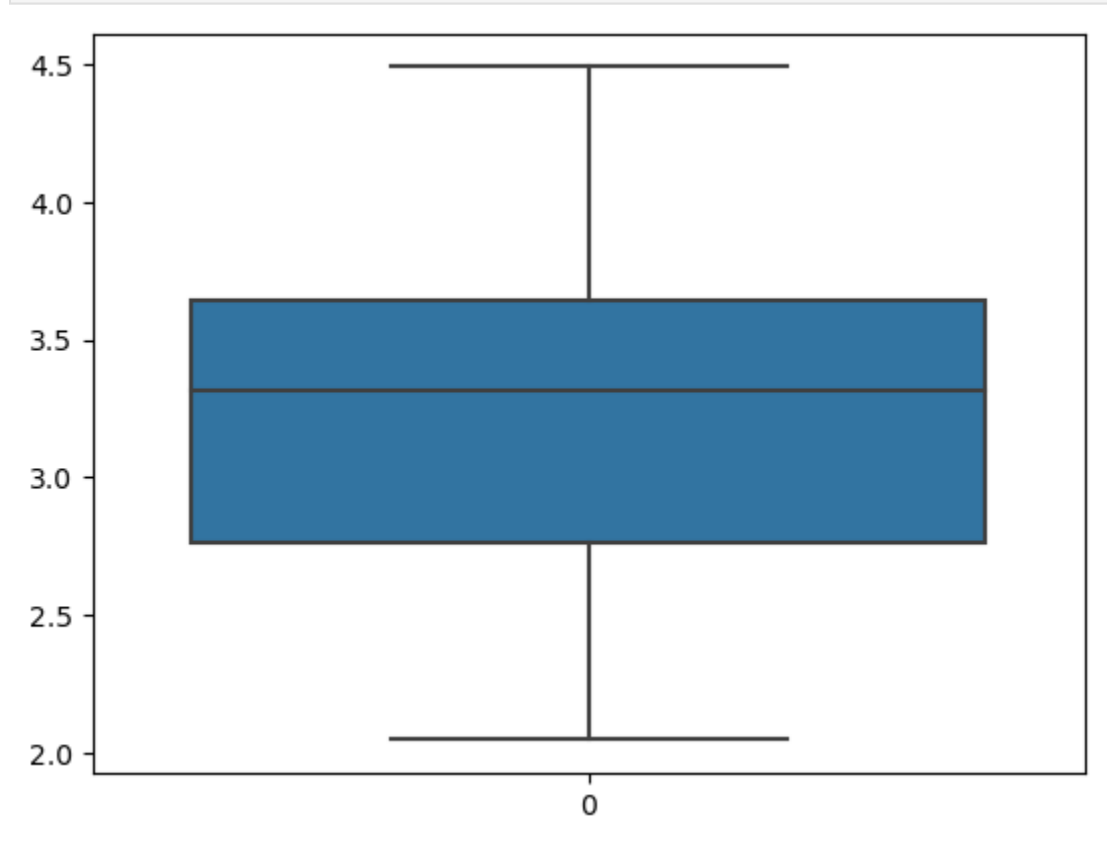
In [15]:
```python
#boxplot:
sns.boxplot(df['Budget'])
```

Out[15]:
```
<Axes: >
```



In [20]:
```python
#Dropping the outliers from Budget_usd:
df=df.drop(df[df['Budget'] >4.5].index)
df=df.drop(df[df['Budget']<2].index)
```

In [21]:
```python
#boxplot after removing outliers:
sns.boxplot(df['Budget'])
plt.show()
```



KMeans

In [22]:
```python
#display  first five records of dataset:
df.head()
```

Out[22]:

	Title	Category Name	Experience	Sub Category Name	Currency	Budget	Location	Freelancer Preferred From	Type	Date Posted	Description	Client Registration Date	Client City	Client Country	Client Currency
0	969	1	0	42	0	3.159532	1	1	0	9407	10434	163	489	61	0
1	6377	7	0	45	1	2.524321	1	1	0	9406	1247	1441	940	129	1
2	1108	0	0	37	1	2.217961	1	1	0	9405	2179	3144	940	129	1
3	467	0	0	90	1	2.310435	1	1	0	9404	2181	3144	940	129	1
5	1818	6	1	26	0	4.199095	1	1	0	9402	568	635	488	128	0

In [23]:
```python
#selecting column for clustering:

#coulmn indexing use to save category and subcategory name in variable x:
x = df.iloc[:, [1,3]]

#display first five records:
x.head()
```

Out[23]:

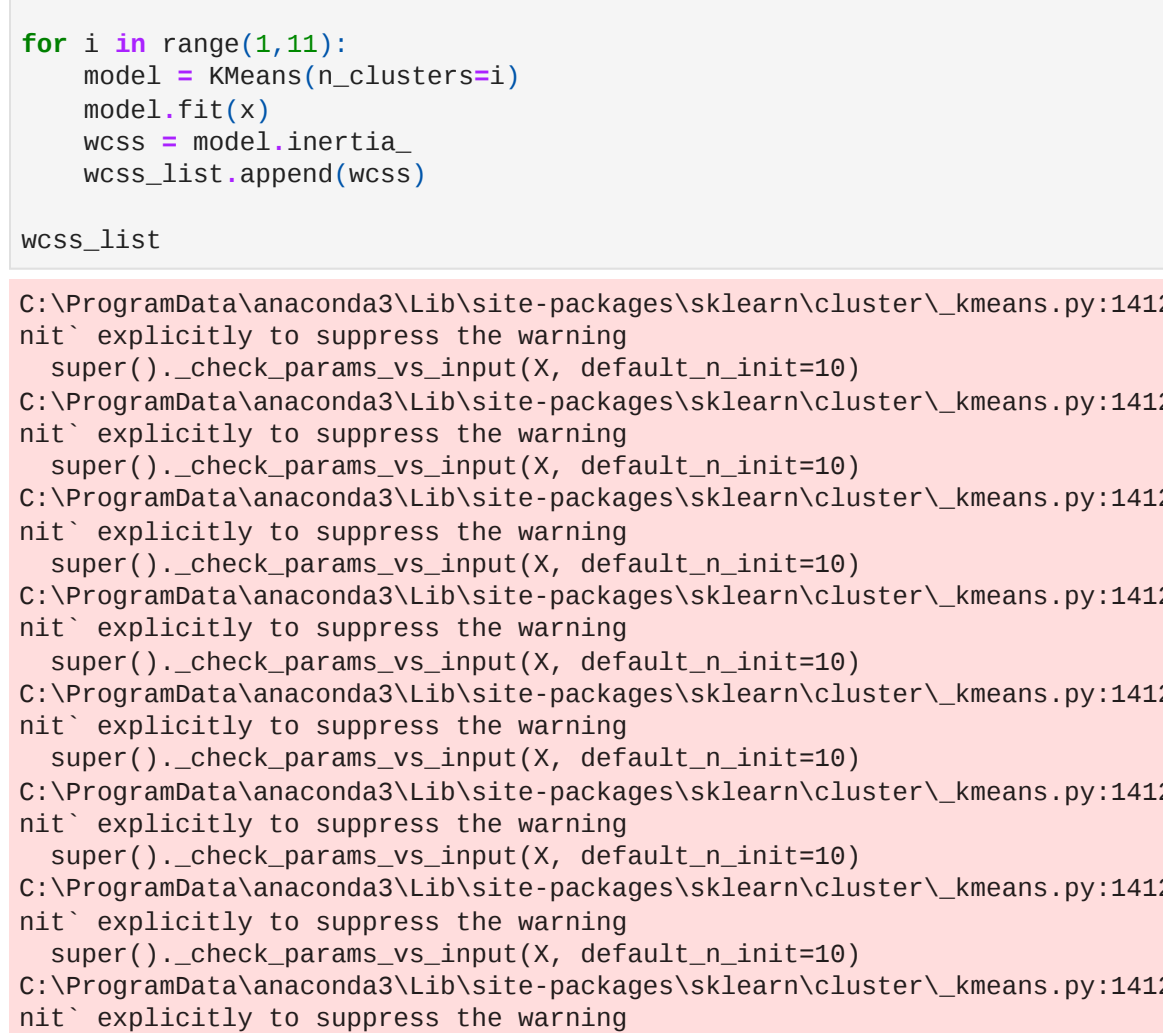
	Category Name	Sub Category Name
0	1	42
1	7	45
2	0	37
3	0	90
5	6	26

In [24]:
```python
#using KMeans Method:
from sklearn.cluster import KMeans

wcss_list = []

for i in range(1,11):
    model = KMeans(n_clusters=i)
    model.fit(x)
    wcss = model.inertia_
    wcss_list.append(wcss)

wcss_list
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

Out[24]:
```
[12626877.477351764,
 2955250.2400581203,
 1376361.7485217797,
 629680.1173443565,
 377634.836149316,
 272008.07151177706,
 204156.69936097844,
 163607.63093468035,
 132979.15309719928,
 109260.27320295836]
```

In [25]:
```python
#plot Elbow method graph:
import matplotlib.pyplot as plt

plt.plot(range(1,11), wcss_list)
plt.title('Elbow Method Graph')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```

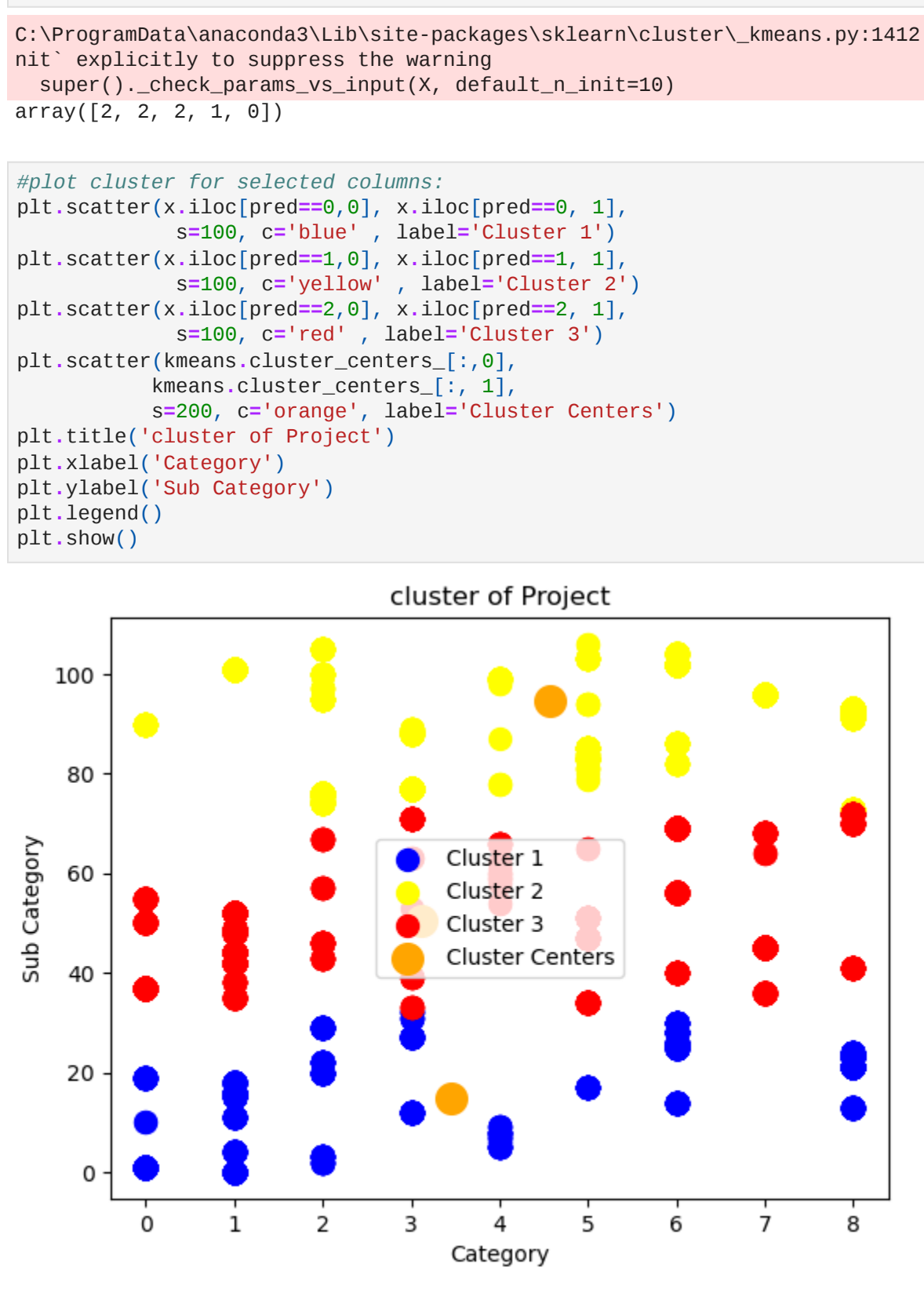


In [26]:
```python
#Here Elbow point is 3:
kmeans = KMeans(n_clusters=3)
kmeans.fit(x)
pred = kmeans.predict(x)
pred[:5]
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

Out[26]:
```
array([2, 2, 2, 1, 0])
```

In [31]:
```python
#plot cluster for selected columns:
plt.scatter(x.iloc[pred==0,0], x.iloc[pred==0, 1],
             s=100, c='blue' , label='Cluster 1')
plt.scatter(x.iloc[pred==1,0], x.iloc[pred==1, 1],
             s=100, c='yellow' , label='Cluster 2')
plt.scatter(x.iloc[pred==2,0], x.iloc[pred==2, 1],
             s=100, c='red' , label='Cluster 3')
plt.scatter(kmeans.cluster_centers_[:,0],
           kmeans.cluster_centers_[:, 1],
           s=200, c='orange', label='Cluster Centers')
plt.title('cluster of Project')
plt.xlabel('Category')
plt.ylabel('Sub Category')
plt.legend()
plt.show()
```



In [ ]: