```
import io
from numpy import unique
from numpy import where
from sklearn.cluster import KMeans
from matplotlib import pyplot
import pandas as pd
from google.colab import files
```

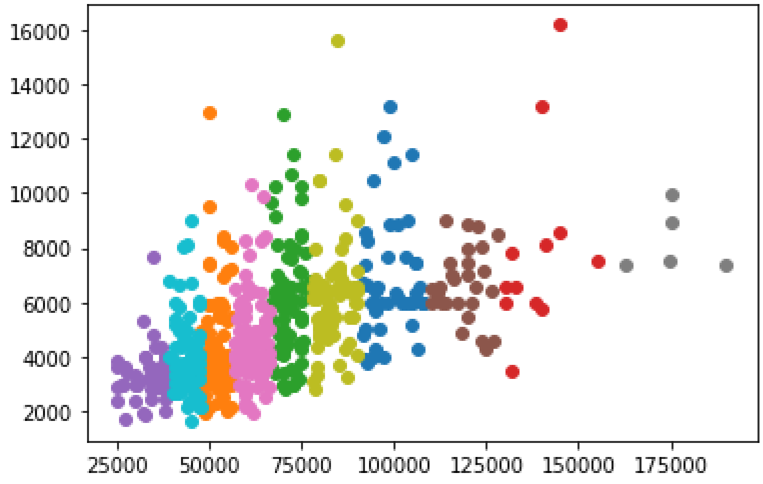
```
uploaded = files.upload()
data = pd.read_csv(io.BytesIO(uploaded['HousePrices canada.csv']))
```

Choose Files HousePrices canada.csv

- **HousePrices canada.csv**(application/vnd.ms-excel) - 24093 bytes, last modified: 5/10/2021 - 100% done

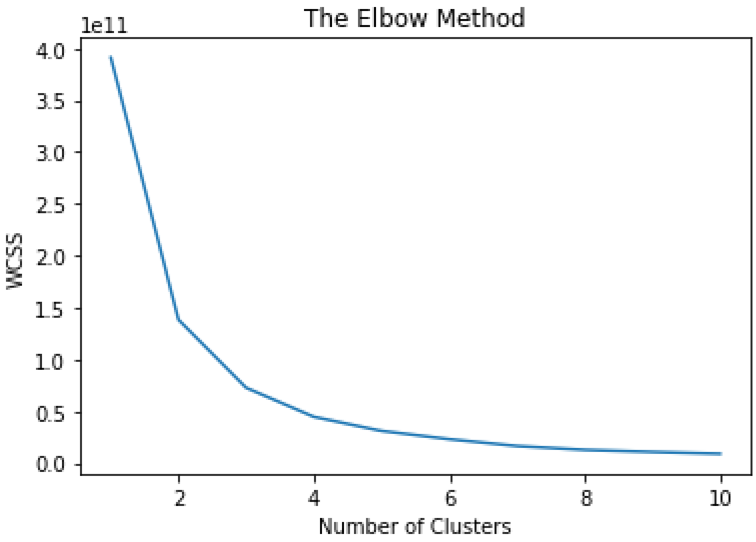
```
Saving HousePrices canada.csv to HousePrices canada (1).csv
```

```
X=data[["price","lotsize"]]
X=(X.to_numpy())
model = KMeans(n_clusters=10)
model.fit(X)
yhat = model.predict(X)
clusters = unique(yhat)
for cluster in clusters:
 row_ix = where(yhat == cluster)
 pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
pyplot.show()
```

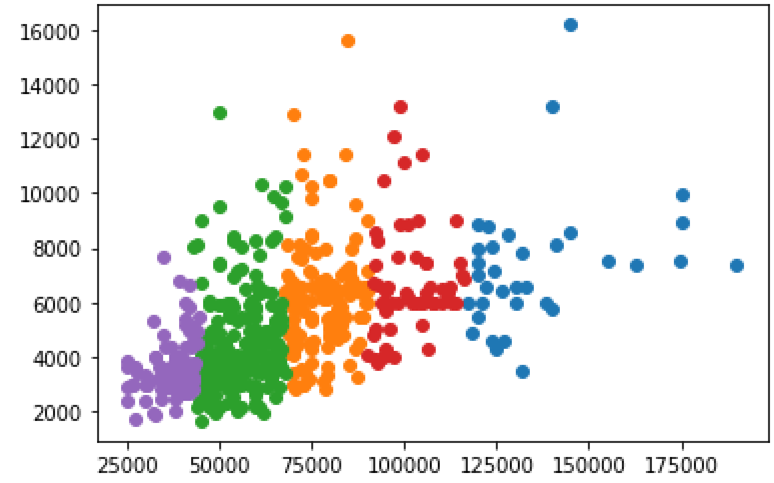


```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
wcss=[]
for i in range(1,11):
 kmeans=KMeans(n_clusters=i, init='k-means++',random_state=0)
 kmeans.fit(X)
 wcss.append(kmeans.inertia_)
```

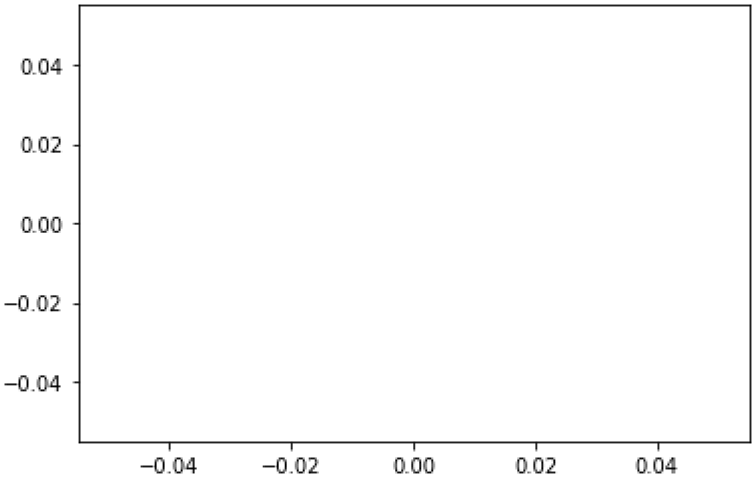
```
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS') #Within Cluster Sum of Squares
plt.show()
```



```
from numpy import unique
from numpy import where
from sklearn.cluster import AgglomerativeClustering
from matplotlib import pyplot
import pandas as pd
data = pd.read_csv("HousePrices canada.csv")
X=data[["price","lotsize"]]
X=(X.to_numpy())
model = AgglomerativeClustering(n_clusters=5)
yhat = model.fit_predict(X)
clusters = unique(yhat)
for cluster in clusters:
 row_ix = where(yhat == cluster)
 pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
pyplot.show()
```



```
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import DBSCAN
from matplotlib import pyplot
data = pd.read_csv("HousePrices canada.csv")
X=data[["price","lotsize"]]
X=(X.to_numpy())
model = DBSCAN(eps=0.30, min_samples=9)
yhat = model.fit_predict(X)
for cluster in clusters:
 row_ix = where(yhat == cluster)
 pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
pyplot.show()
```



✓ 0s completed at 22:51 ● ×