```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from numpy.ma.core import nonzero
linear=LinearRegression()
linear.fit(xtrain,ytrain)
print("intercept is",linear.intercept_,"coef is",linear.coef_)
```

```
intercept is -808214.8941724668 coef is [682.979958]
```

```
ypred=linear.predict(xtest)
print("predicted value",ypred)
```

```
predicted value [557745.0218229  548183.30241093 561159.92161289 ... 546134.36253694
 542036.48278895 546817.34249494]
```

```
rsquare=linear.score(xtest,ytest)
print("r square value",rsquare)
```

```
r square value 0.001334799698434308
```

```
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(ytest,ypred)
rmse=mean_squared_error(ytest,ypred,squared=False)
print("mse=",mse)
print("rmse",rmse)
print("r_Squared=",rsquare)
```

```
mse= 172370583344.17865
rmse 415175.36456800834
r_Squared= 0.001334799698434308
```

Tast 2 **K-Means Clustring**(Mall_customers DataSet)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('/content/drive/MyDrive/Mall_Customers.csv')
df.head()
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Next steps: Generate code with df | View recommended plots

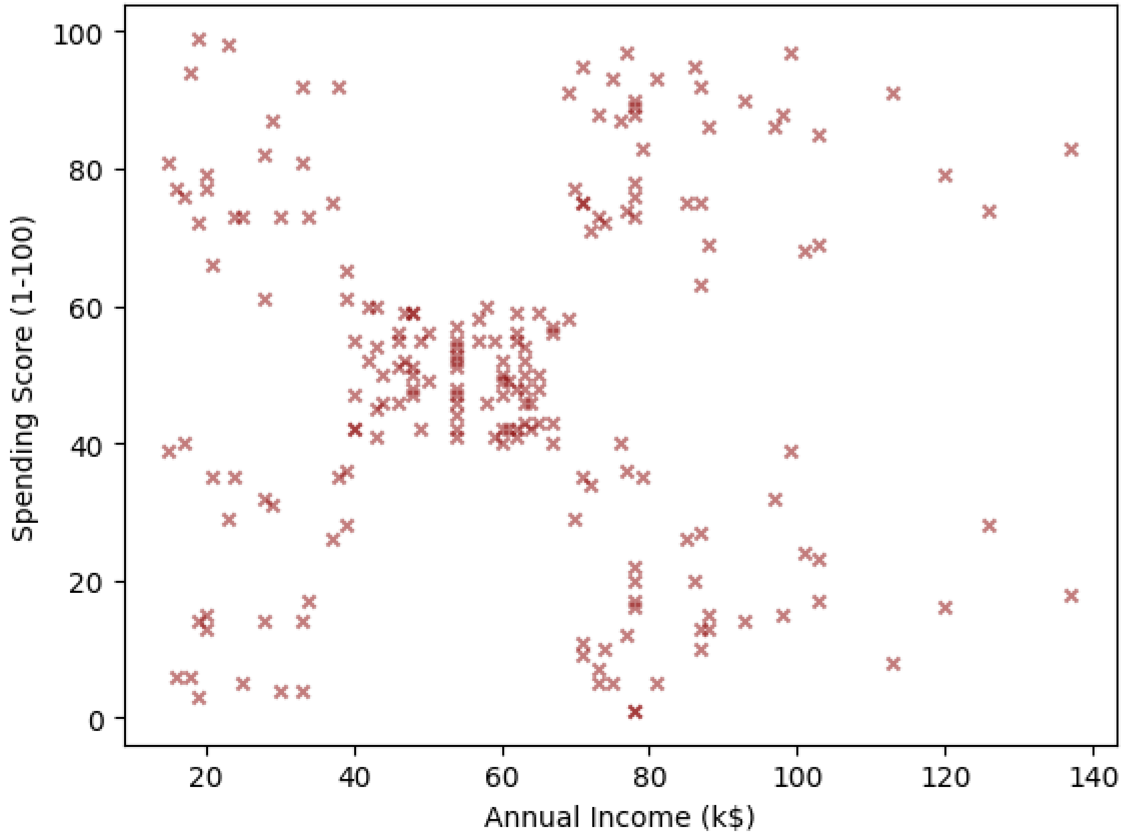
```
from sklearn import preprocessing
label_encoder=preprocessing.LabelEncoder()
df['Gender']=label_encoder.fit_transform(df['Gender'])
df.head()
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40

Next steps: Generate code with df | View recommended plots

```
X_train =df[["Annual Income (k$)", "Spending Score (1-100)"]]
X_train.plot.scatter(x="Annual Income (k$)", y="Spending Score (1-100)",
                     c="darkred", marker="x", alpha=.5)
```

```
<Axes: xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



Based on the scatter plot we divide the data into 5 different clusters

```
from sklearn.cluster import KMeans
model = KMeans(n_clusters=5)
model.fit(X_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
▾       KMeans
KMeans(n_clusters=5)
```

```
# Extract the centroids and the clusters.
centroids = model.cluster_centers_
clusters = model.labels_
print(centroids)
print(clusters)
```

```
[[55.2962963  49.51851852]
 [86.53846154 82.12820513]
 [88.2        17.11428571]
 [25.72727273 79.36363636]
 [26.30434783 20.91304348]]
[4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
```

```
 3 4 3 4 3 4 0 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 0 1 2 1 2 1 0 1 2 1 2 1 2 1 2 1 0 1 2 1 2 1
 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1]
```

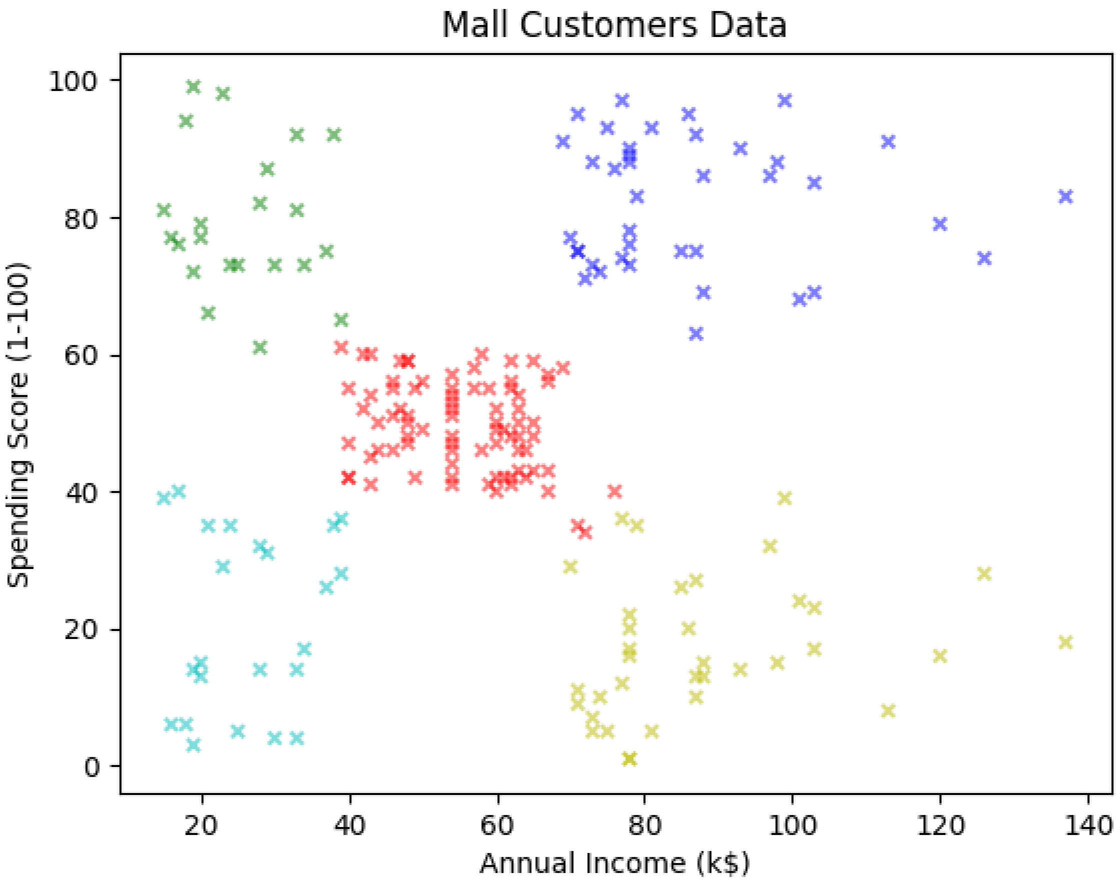
```
clusters = pd.Series(clusters).map({
    0: "r",
    1: "b",
    2: "y",
    3: "g",
    4: "c",
},na_action='ignore')
```

```
print(clusters)
```

```
0      c
1      g
2      c
3      g
4      c
      ..
195    b
196    y
197    b
198    y
199    b
Length: 200, dtype: object
```

```
X_train.plot.scatter(x="Annual Income (k$)", y="Spending Score (1-100)",
                     c=clusters, marker="x", alpha=.5, title="Mall Customers Data")
```

```
<Axes: title={'center': 'Mall Customers Data'}, xlabel='Annual Income (k$)',
ylabel='Spending Score (1-100)'>
```



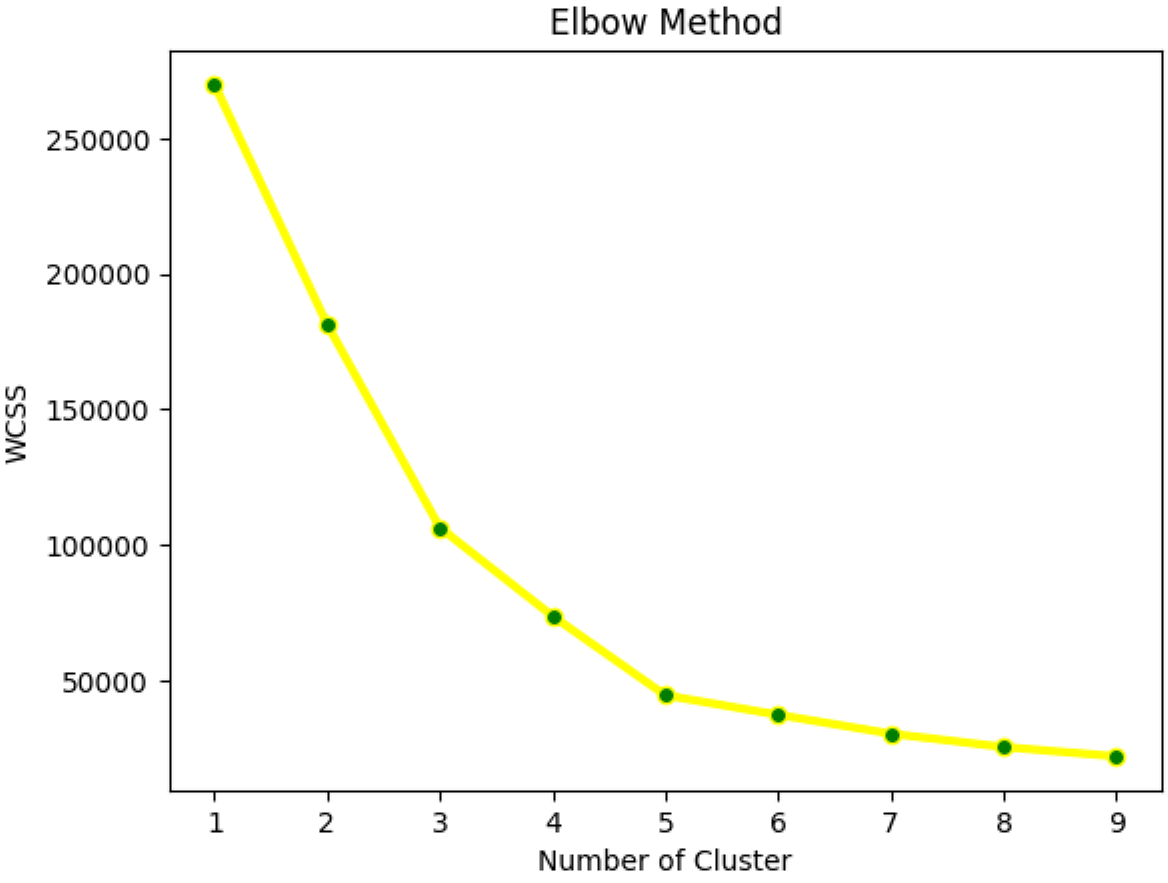
```
from sklearn.metrics import silhouette_score
from scipy.spatial import distance # To calculate distances

sse = {}
for k in range(1,10):
    model = KMeans(n_clusters=k)
    model.fit(X_train)
    # y_kmeans = kmeans.predict(X)
    sse[k] = model.inertia_
    print("For cluster = {}, WCSS is {}".format(k, sse[k]))

plt.figure()
plt.plot(list(sse.keys()),list(sse.values()),linewidth=3, color = 'Yellow', marker='o', mark
plt.xlabel("Number of Cluster")
plt.ylabel("WCSS")
plt.title("Elbow Method")
plt.show()

print('Optimum Number of Cluster : 5')
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
For cluster = 1, WCSS is 269981.28
For cluster = 2, WCSS is 181363.59595959593
For cluster = 3, WCSS is 106348.37306211122
For cluster = 4, WCSS is 73679.78903948836
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
For cluster = 5, WCSS is 44448.4554479337
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
For cluster = 6, WCSS is 37233.814510710006
For cluster = 7, WCSS is 30273.394312070042
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
For cluster = 8, WCSS is 25354.360937251142
For cluster = 9, WCSS is 21900.341350107527
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: T
  warnings.warn(
```



```
Optimum Number of Cluster : 5
```

```
ss = {}

for k in range(2,10):
    model = KMeans(n_clusters=k)
    model.fit(X_train)
    label = model.labels_
    sil_coeff = silhouette_score(X_train,label,metric = 'euclidean')
    ss[k] = sil_coeff
    print('For cluster= {}, Silhouette Coefficient is {}'.format(k,sil_coeff))

plt.figure()
plt.plot(list(ss.keys()),list(ss.values()), linewidth=3, color = 'Yellow', marker='o', marker
plt.xlabel("Number of Cluster")
plt.ylabel("Silhouette Score")
plt.title("Silhouette Analysis")
plt.show()

print('Optimum Number of Cluster : 5')
```

```
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
 warnings.warn(
or cluster= 2, Silhouette Coefficient is 0.2968969162503008
or cluster= 3, Silhouette Coefficient is 0.46761358158775435
or cluster= 4, Silhouette Coefficient is 0.4931963109249047
or cluster= 5, Silhouette Coefficient is 0.553931997444648
```