

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

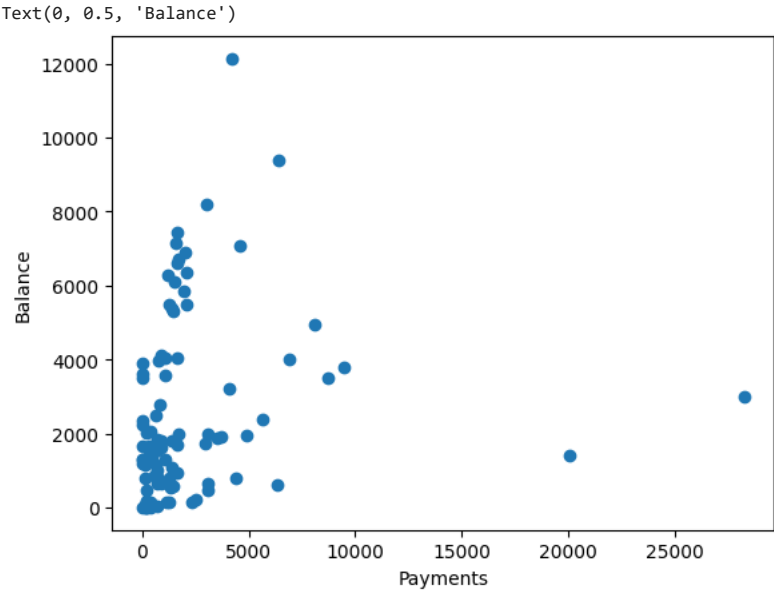
```
df = pd.read_csv('/CC GENERAL.csv')
df.head()
```

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY
0	C10001	40.900749	0.818182	95.40	0.00	95.4	0.000000	0.166667
1	C10002	3202.467416	0.909091	0.00	0.00	0.0	6442.945483	0.000000
2	C10003	2495.148862	1.000000	773.17	773.17	0.0	0.000000	1.000000
3	C10004	1666.670542	0.636364	1499.00	1499.00	0.0	205.788017	0.083333
4	C10005	817.714335	1.000000	16.00	16.00	0.0	0.000000	0.083333

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 18 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   CUST_ID                                   100 non-null    object
1   BALANCE                                  100 non-null    float64
2   BALANCE_FREQUENCY                       100 non-null    float64
3   PURCHASES                               100 non-null    float64
4   ONEOFF_PURCHASES                        100 non-null    float64
5   INSTALLMENTS_PURCHASES                  100 non-null    float64
6   CASH_ADVANCE                             100 non-null    float64
7   PURCHASES_FREQUENCY                     100 non-null    float64
8   ONEOFF_PURCHASES_FREQUENCY              100 non-null    float64
9   PURCHASES_INSTALLMENTS_FREQUENCY        100 non-null    float64
10  CASH_ADVANCE_FREQUENCY                  100 non-null    float64
11  CASH_ADVANCE_TRX                        100 non-null    int64
12  PURCHASES_TRX                           100 non-null    int64
13  CREDIT_LIMIT                            100 non-null    int64
14  PAYMENTS                                 100 non-null    float64
15  MINIMUM_PAYMENTS                        89 non-null     float64
16  PRC_FULL_PAYMENT                        100 non-null    float64
17  TENURE                                  100 non-null    int64
dtypes: float64(13), int64(4), object(1)
memory usage: 14.2+ KB
```

```
plt.scatter(df.PAYMENTS,df.BALANCE)
plt.xlabel('Payments')
plt.ylabel('Balance')
```



```
km = KMeans(n_clusters=4)
y_predicted = km.fit_predict(df[['PAYMENTS','BALANCE']])
```

y_predicted

```

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
warnings.warn(
array([[2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3,
        2, 0, 3, 2, 2, 2, 3, 2, 3, 2, 3, 2, 0, 2, 3, 2, 3, 1, 2, 2, 2, 2,
        3, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0, 2, 3, 0, 2, 0, 2, 0, 2,
        2, 2, 0, 2, 3, 1, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 0, 2, 3, 3,
        2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int32)

```

```

df['cluster']=y_predicted
df.head()

```

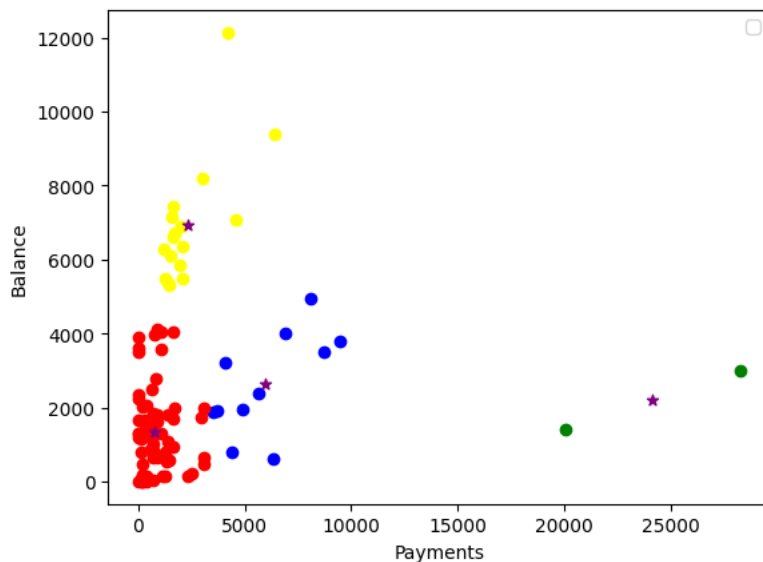
	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY
0	C10001	40.900749	0.818182	95.40	0.00	95.4	0.000000	0.166667
1	C10002	3202.467416	0.909091	0.00	0.00	0.0	6442.945483	0.000000
2	C10003	2495.148862	1.000000	773.17	773.17	0.0	0.000000	1.000000
3	C10004	1666.670542	0.636364	1499.00	1499.00	0.0	205.788017	0.083333
4	C10005	817.714335	1.000000	16.00	16.00	0.0	0.000000	0.083333

```

df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
df4 = df[df.cluster==3]
plt.scatter(df1.PAYMENTS,df1.BALANCE,color='blue');
plt.scatter(df2.PAYMENTS,df2.BALANCE,color='green');
plt.scatter(df3.PAYMENTS,df3.BALANCE,color='red');
plt.scatter(df4.PAYMENTS,df4.BALANCE,color='yellow');
plt.scatter(km.cluster_centers_[0,0],km.cluster_centers_[0,1],color='purple',marker='*')
plt.xlabel('Payments')
plt.ylabel('Balance')
plt.legend()

```

WARNING:matplotlib.legend.No artists with labels found to put in legend. Note that
<matplotlib.legend.Legend at 0x7d745f012a40>



```

scaler = MinMaxScaler()
scaler.fit(df[['BALANCE']])
df['BALANCE'] = scaler.transform(df[['BALANCE']])
scaler.fit(df[['PAYMENTS']])
df['PAYMENTS'] = scaler.transform(df[['PAYMENTS']])

```

```

sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['PAYMENTS', 'BALANCE']])
    sse.append(km.inertia_)

```

◀ ▶

```
[<matplotlib.lines.Line2D at 0x7d745a67e200>]
```

