```
from google.colab import files
uploaded=files.upload()
```

```
Choose files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving loan_data.csv to loan_data.csv
```

```
import pandas as pd
import seaborn as sns
```

```
d1=pd.read_csv('loan_data.csv')
d1
```

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revo
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	
2	1	debt_consolidation	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	
3	1	debt_consolidation	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	
4	1	credit_card	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	
...	...	...	...	...	...	...	...	...	...	
9573	0	all_other	0.1461	344.76	12.180755	10.39	672	10474.000000	215372	
9574	0	all_other	0.1253	257.70	11.141862	0.21	722	4380.000000	184	
9575	0	debt_consolidation	0.1071	97.81	10.596635	13.09	687	3450.041667	10036	
9576	0	home_improvement	0.1600	351.58	10.819778	19.18	692	1800.000000	0	
9577	0	debt_consolidation	0.1392	853.43	11.264464	16.28	732	4740.000000	37879	

9578 rows × 14 columns

```
d1.info()
```

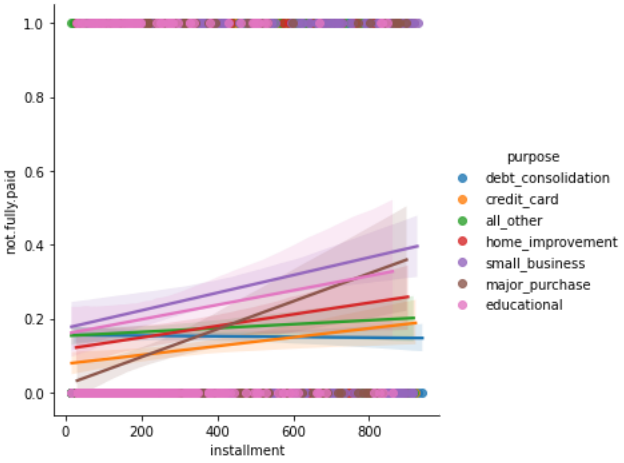
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   credit.policy      9578 non-null   int64  
 1   purpose            9578 non-null   object 
 2   int.rate           9578 non-null   float64
 3   installment        9578 non-null   float64
 4   log.annual.inc     9578 non-null   float64
 5   dti                9578 non-null   float64
 6   fico               9578 non-null   int64  
 7   days.with.cr.line  9578 non-null   float64
 8   revol.bal          9578 non-null   int64  
 9   revol.util         9578 non-null   float64
 10  inq.last.6mths     9578 non-null   int64  
 11  delinq.2yrs        9578 non-null   int64  
 12  pub.rec            9578 non-null   int64  
 13  not.fully.paid     9578 non-null   int64  
dtypes: float64(6), int64(7), object(1)
memory usage: 1.0+ MB
```

```
sns.lmplot('installment','int.rate',hue='purpose',data=d1)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword
  FutureWarning
<seaborn.axisgrid.FacetGrid at 0x7f1d5691f910>
```


```
sns.lmplot('installment','not.fully.paid',hue='purpose',data=d1)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnir
  FutureWarning
<seaborn.axisgrid.FacetGrid at 0x7f1d53f23110>
```




```
from sklearn.cluster import KMeans
kmeans=KMeans(n_clusters=7)
```

Fit the Model to all the Data Except for Purpose Segment

```
kmeans.fit(d1.drop('purpose',axis=1))
```

```
KMeans(n_clusters=7)
```

**In This Case Total 7 Different Varients are available, Every Varient can be considered as unique cluster.cluster means one kind of geometrical shape. This Can be Any Shape by nature. But Every geometrical shape must have one centre , in case of preparing cluster (one shape) we need to find first centre value then only we can draw any shape. **

```
centers=kmeans.cluster_centers_
centers
```

```
array([[ 8.53338796e-01,  1.26959156e-01,  3.97439873e+02,
         1.11387068e+01,  1.52012331e+01,  7.06258501e+02,
         5.19242389e+03,  2.31148947e+04,  6.18544449e+01,
         1.47357640e+00,  1.33142155e-01,  4.42441622e-02,
         1.59770586e-01],
       [ 0.00000000e+00,  1.75350000e-01,  7.83290000e+02,
         1.30351867e+01,  1.28850000e+01,  6.94500000e+02,
         8.07002083e+03,  1.07968600e+06,  7.75000000e+01,
         6.50000000e+00,  0.00000000e+00,  0.00000000e+00,
         5.00000000e-01],
       [-5.55111512e-16,  1.31265672e-01,  5.66934925e+02,
         1.20030768e+01,  1.72144776e+01,  7.21477612e+02,
         7.40640672e+03,  2.25105299e+05,  5.28805970e+01,
         2.11940299e+00,  1.34328358e-01,  2.98507463e-02,
         3.13432836e-01],
       [ 8.10471861e-01,  1.20288227e-01,  2.74200194e+02,
         1.07699992e+01,  1.11616429e+01,  7.12496159e+02,
         4.10240691e+03,  5.36568788e+03,  3.92754099e+01,
         1.60605111e+00,  1.82316978e-01,  7.25819094e-02,
         1.53472331e-01],
       [ 0.00000000e+00,  1.37810000e-01,  5.74971000e+02,
         1.24698456e+01,  1.51920000e+01,  7.07000000e+02,
         8.09491250e+03,  4.15457800e+05,  5.99500000e+01,
         1.60000000e+00,  1.00000000e-01,  1.00000000e-01,
         5.00000000e-01],
       [ 7.49084249e-01,  1.28048352e-01,  4.24326996e+02,
         1.15105992e+01,  1.63994322e+01,  7.10131868e+02,
         6.17668872e+03,  6.02746502e+04,  6.32780220e+01,
         1.55860806e+00,  9.89010989e-02,  3.47985348e-02,
         1.92307692e-01],
       [ 3.60902256e-01,  1.27688722e-01,  4.50986165e+02,
         1.18544918e+01,  1.62051880e+01,  7.14030075e+02,
         6.56599499e+03,  1.28156398e+05,  5.91822556e+01,
```

```
1.84210526e+00,  1.20300752e-01,  1.50375940e-02,
2.40601504e-01]])
```

Already You have fitted model, get the labels for kmeans and create one new column with the name of klabels or any name which a developer can understand

```
d1['klabels']=kmeans.labels_
d1
```

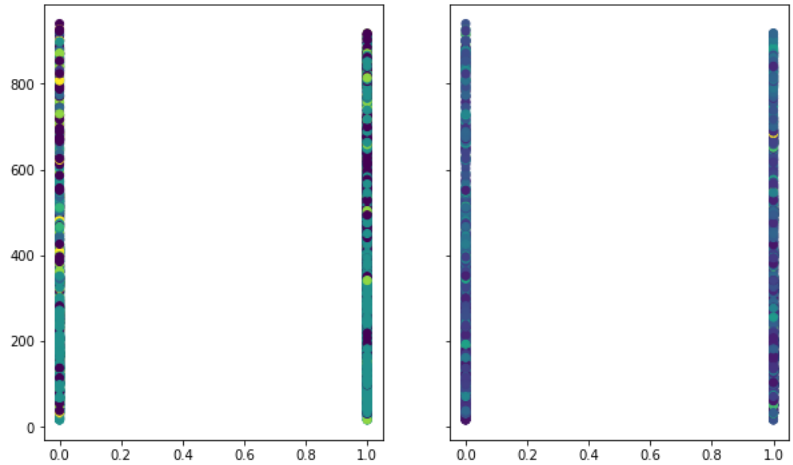
	credit.policy	purpose	int.rate	installment	log.annual.inc	d
0	1	debt_consolidation	0.1189	829.10	11.350407	19.
1	1	credit_card	0.1071	228.22	11.082143	14.
2	1	debt_consolidation	0.1357	366.86	10.373491	11.
3	1	debt_consolidation	0.1008	162.34	11.350407	8.
4	1	credit_card	0.1426	102.92	11.299732	14.
...	...	...	...	...	...	
9573	0	all_other	0.1461	344.76	12.180755	10.
9574	0	all_other	0.1253	257.70	11.141862	0.
9575	0	debt_consolidation	0.1071	97.81	10.596635	13.
9576	0	home_improvement	0.1600	351.58	10.819778	19.
9577	0	debt_consolidation	0.1392	853.43	11.264464	16.

9578 rows × 15 columns

```
import matplotlib.pyplot as plt
```

```
f,(ax1,ax2)=plt.subplots(nrows=1,ncols=2,sharey=True,figsize=(10,6))
# Let's Draw Graph for With Cluster Concept
ax1.scatter(x=d1['credit.policy'],y=d1['installment'],c=d1['klabels'])
#Let's Draw the Graph withut Cluster
ax2.scatter(x=d1['credit.policy'],y=d1['installment'],c=d1['days.with.cr.line'])
```

```
<matplotlib.collections.PathCollection at 0x7f1d4dcd0290>
```




```
from google.colab import files
uploaded=files.upload()
```

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Seed_Data.csv to Seed_Data.csv
```

```
d2=pd.read_csv('Seed_Data.csv')
d2
```

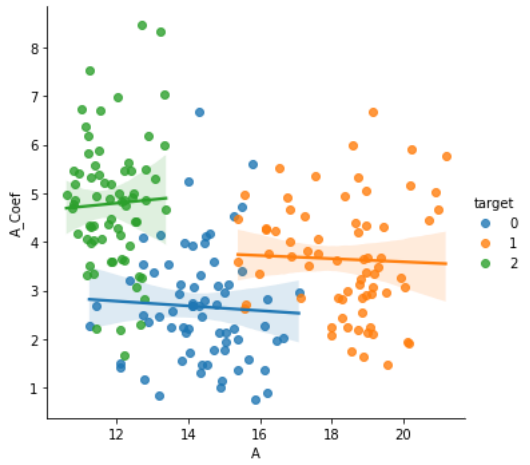
	A	P	C	LK	WK	A_Coef	LKG	target
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	0
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	0
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	0
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	0
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	0
...	...	...	...	...	...	...	...	...
205	12.19	13.20	0.8783	5.137	2.981	3.631	4.870	2
206	11.23	12.88	0.8511	5.140	2.795	4.325	5.003	2

```
d2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   A       210 non-null    float64
 1   P       210 non-null    float64
 2   C       210 non-null    float64
 3   LK      210 non-null    float64
 4   WK      210 non-null    float64
 5   A_Coef  210 non-null    float64
 6   LKG     210 non-null    float64
 7   target  210 non-null    int64  
dtypes: float64(7), int64(1)
memory usage: 13.2 KB
```

```
sns.lmplot('A','A_Coef',data=d2,hue='target')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnir
  FutureWarning
<seaborn.axisgrid.FacetGrid at 0x7f1d4dd82150>
```




```
from sklearn.cluster import KMeans
kmeans1=KMeans(n_clusters=3)
```

```
kmeans1.fit(d2.drop('target',axis=1))
```

```
KMeans(n_clusters=3)
```

```
d2['klabels1']=kmeans1.labels_
d2.head()
```

	A	P	C	LK	WK	A_Coef	LKG	target	klabels1
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	0	2
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	0	2
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	0	2
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	0	2
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	0	2

```
f,(ax1,ax2)=plt.subplots(nrows=1,ncols=2,sharey=True,figsize=(10,6))
#with Cluster
ax1.scatter(x=d2['A'],y=d2['A_Coef'],c=d2['klabels1'])
#Without Cluster
ax2.scatter(x=d2['A'],y=d2['A_Coef'],c=d2['target'])
```

```
<matplotlib.collections.PathCollection at 0x7f1d4d9f62d0>
```