Experiment 11

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

```
# Suppressing Warnings import
warnings
warnings.filterwarnings
```

```
# Importing Pandas and NumPy import
pandas as pd, numpy as np import
matplotlib.pyplot as plt import seaborn
as sns
```

```
path1 = "/content/drive/MyDrive/Machine Learning for Real world Application (PCC-
CSD601 and PCC-CSD691)/Laboratory/Exp 12/College_Data.csv"
df = pd.read_csv(path1)
```

```
df.head(3)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 777,\n  \"fields\": [\ n     {\n
      \"column\": \"Unnamed: 0\",\n                          \"properties\": {\n
\"dtype\": \"string\",\n                \"num_unique_values\": 777,\n
\"samples\": [\n                  \"SUNY College at Fredonia\",\n
\"Southeast Missouri State University\",\n \"University         of
Cincinnati\"\n              ],\n           \"semantic_type\": \"\",\n
\"description\": \"\"\n             }\n      },\n      {\n         \"column\":
\"Private\",\n           \"properties\": {\n               \"dtype\":
\"category\",\n \"num_unique_values\": 2,\n                  \"samples\": [\n \"No\",\n
                          \"Yes\"\n  ],\n
\"semantic_type\": \"\",\n                \"description\": \"\"\n             }\
n      },\n      {\n         \"column\": \"Apps\",\n              \"properties\": {\n
\"dtype\": \"number\",\n                \"std\": 3870,\n              \"min\": 81,\n
\"max\": 48094,\n              \"num_unique_values\": 711,\n
\"samples\": [\n                323,\n              1127\n            ],\n
\"semantic_type\": \"\",\n                \"description\": \"\"\n             }\
n      },\n      {\n         \"column\": \"Accept\",\n             \"properties\":
{\n           \"dtype\": \"number\",\n                \"std\": 2451,\n
\"min\": 72,\n    \"max\": 26330,\n                          \"num_unique_values\":
693,\n           \"samples\": [\n                689,\n              1623\
n          ],\n           \"semantic_type\": \"\",\n
\"description\": \"\"\n             }\n      },\n      {\n         \"column\":
\"Enroll\",\n           \"properties\": {\n              \"dtype\": \"number\",\n
\"std\": 929,\n              \"min\": 35,\n              \"max\": 6392,\n
\"num_unique_values\": 581,\n  \"samples\": [\n                                  769,\n
223\n           ],\n           \"semantic_type\": \"\",\n
\"description\": \"\"\n             }\n      },\n      {\n         \"column\":
```

{\n   \"dtype\": \"number\",\n   \"std\": 14,\n
\"min\": 24,\n \"max\": 100,\n   \"num_unique_values\": 65,\n
                \"samples\": [\n   43,\n   54\n   ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"S.F.Ratio\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
3.958349135205549,\n   \"min\": 2.5,\n   \"max\": 39.8,\n
\"num_unique_values\": 173,\n   \"samples\": [\n   25.3,\n
14.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"perc.alumni\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 12,\n   \"min\": 0,\n
\"max\": 64,\n \"num_unique_values\": 61,\n   \"samples\": [\n
                12,\n   11\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"Expend\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 5221,\n   \"min\": 3186,\n
\"max\": 56233,\n   \"num_unique_values\": 744,\n
\"samples\": [\n   17500,\n   13705\n   ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Grad.Rate\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
17,\n   \"min\": 10,\n   \"max\": 118,\n
\"num_unique_values\": 81,\n \"samples\": [\n   35,\n
60\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"df"}

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776 Data
columns (total 19 columns):
 #    Column          Non-Null Count  Dtype

-----.
 0    Unnamed: 0       777  non-null       object
 1    Private          777  non-null       object
 2    Apps             777  non-null       int64
 3    Accept           777  non-null       int64
 4    Enroll           777  non-null       int64
 5    Top10perc        777  non-null       int64
 6    Top25perc        777  non-null       int64
 7    F.Undergrad      777  non-null       int64
 8    P.Undergrad      777  non-null       int64
 9    Outstate         777  non-null       int64
 10   Room.Board       777  non-null       int64
 11   Books            777  non-null       int64
 12   Personal         777  non-null       int64
 13   PhD              777  non-null       int64
 14   Terminal         777  non-null       int64
```

```
 15   S.F.Ratio       777 non-null        float64
 16   perc.alumni  777 non-null           int64
 17   Expend          777 non-null        int64
 18   Grad.Rate       777  non-null       int64
dtypes: float64(1), int64(16), object(2) memory
usage: 115.5+ KB
```

```
df.isnull().sum()
```

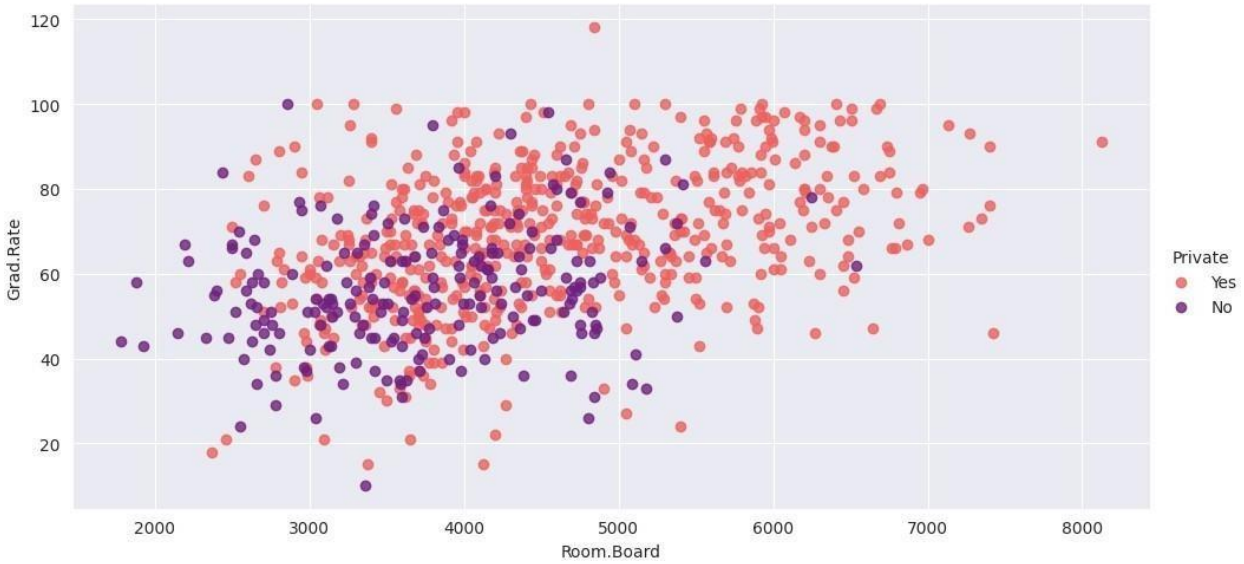
```
Unnamed: 0        0
Private           0
Apps Accept       0
Enroll            0
Top10perc         0
Top25perc         0
F.Undergrad       0
P.Undergrad       0
Outstate          0
Room.Board        0
Books             0
Personal PhD      0
Terminal          0
S.F.Ratio         0
```

```
dtype: int64 df.describe()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n        \"column\": \"Apps\",\n            \"properties\": {\n
\"dtype\": \"number\",\n                \"std\": 16373.62804557302,\n
\"min\": 81.0,\n             \"max\": 48094.0,\n
\"num_unique_values\": 8,\n          \"samples\": [\n
3001.6383526383524,\n            1558.0,\n            777.0\n          ],\n
\"semantic_type\": \"\",\n          \"description\": \"\"\n       }\
{\n          \"dtype\": \"number\",\n               \"std\":
8874.893119771099,\n               \"min\": 72.0,\n              \"max\":   26330.0,\
n          \"num_unique_values\": 8,\n\"samples\":         [\n
2018.8043758043757,\n                1110.0,\n               777.0\n            ],\n
\"semantic_type\": \"\",\n              \"description\": \"\"\n            }\
n     },\n      {\n         \"column\": \"Enroll\",\n             \"properties\":
{\n          \"dtype\": \"number\",\n               \"std\": 2078.352434211878,\n
          \"min\": 35.0,\n     \"max\": 6392.0,\n
\"num_unique_values\": 8,\n                \"samples\": [\n 779.972972972973,\n
                                434.0,\n              777.0\n            ],\n
```

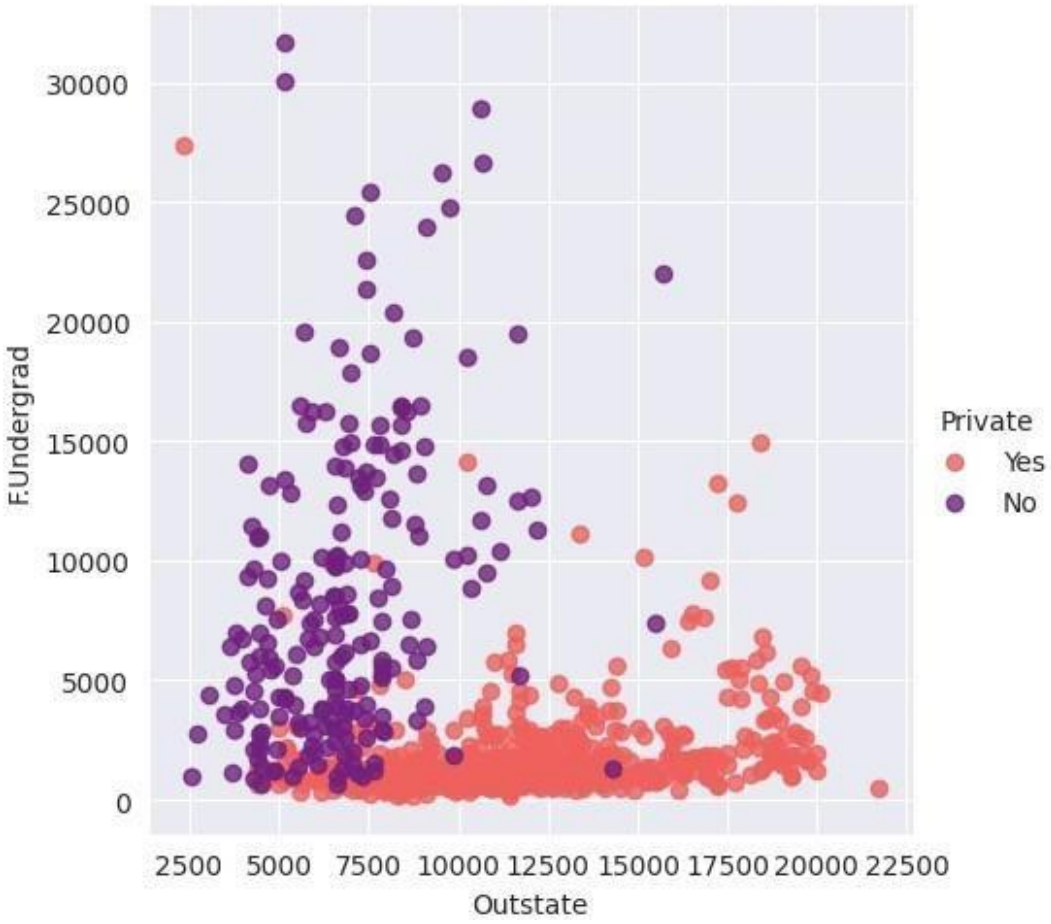
```
sns.set_style('darkgrid')
sns.lmplot(x ='Room.Board',y = 'Grad.Rate',data=df,
hue='Private',palette='magma_r', aspect=2,fit_reg=False)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f6b55722510>
```

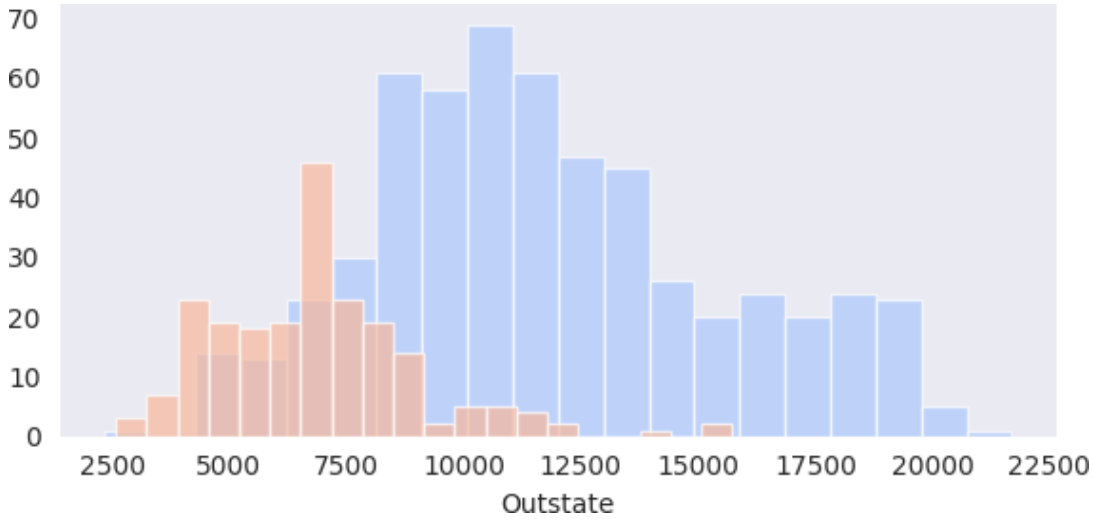



```
sns.set_style('darkgrid')
sns.lmplot(x ='Outstate',y = 'F.Undergrad',data=df,
hue='Private',palette='magma_r',aspect=1,fit_reg=False)
```

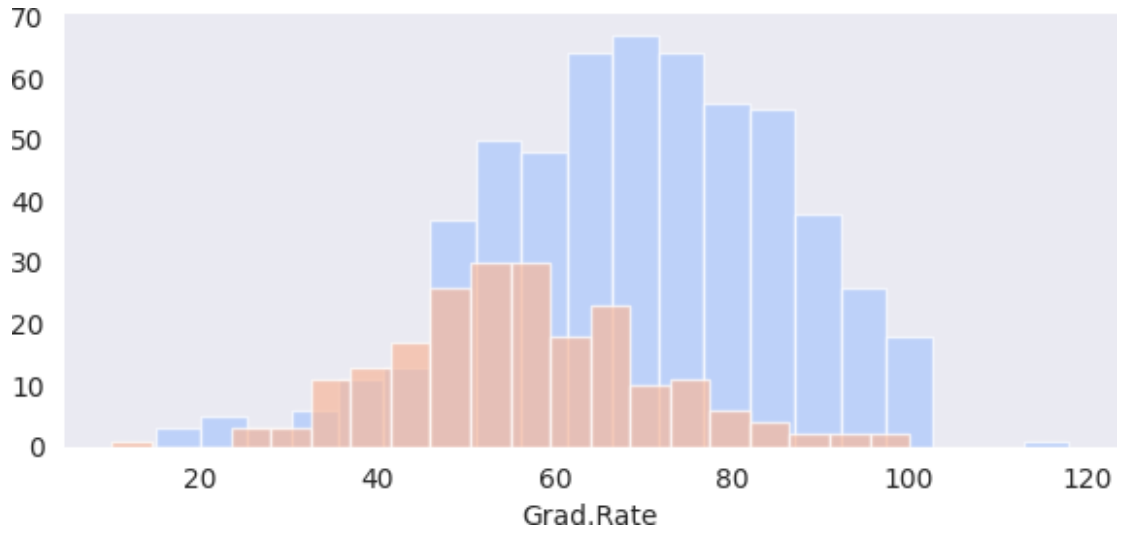
```
<seaborn.axisgrid.FacetGrid at 0x7f6b55aa3c50>
```




```
sns.set_style('dark')
h = sns.FacetGrid(df,hue="Private",palette='coolwarm',aspect=2)
h = h.map(plt.hist,'Outstate',bins=20,alpha=0.7)
```



```
sns.set_style('dark')
g = sns.FacetGrid(df,hue="Private",palette='coolwarm',aspect=2)
g = g.map(plt.hist,'Grad.Rate',bins=20,alpha=0.7)
```

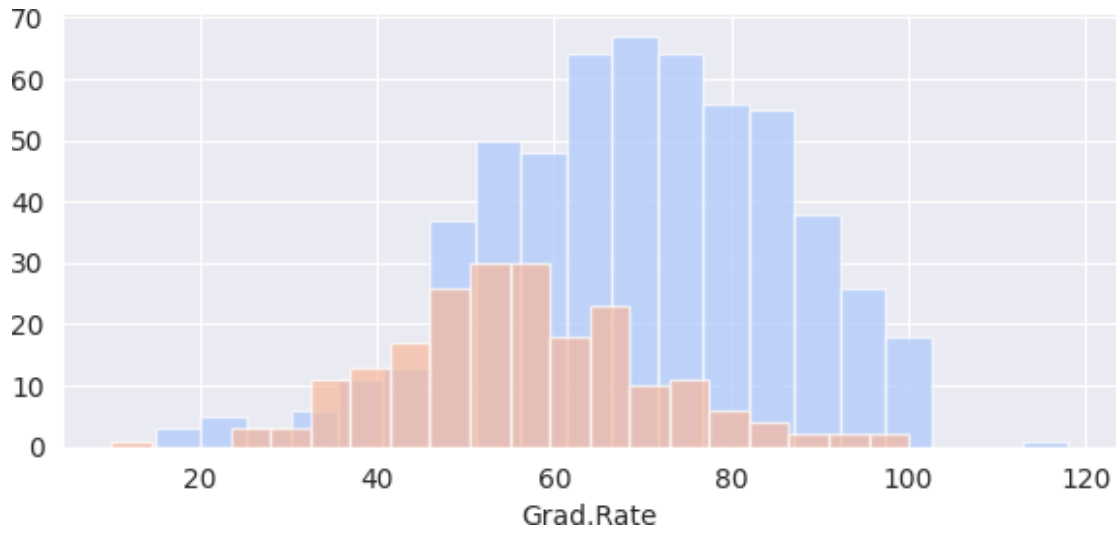



```
df[df['Grad.Rate'] > 100]
```


```
{"summary":"{\n  \"name\": \"df[df['Grad\",\n  \"rows\": 1,\n
\"fields\": [\n         {\n         \"column\": \"Unnamed: 0\",\n
\"properties\": {\n               \"dtype\": \"string\",\n
\"num_unique_values\": 1,\n                \"samples\": [\n
\"Cazenovia College\"\n               ],\n           \"semantic_type\": \"\",\n
\"description\": \"\"\n             }\n      },\n      {\n         \"column\":
\"Private\",\n           \"properties\": {\n              \"dtype\": \"string\",\ n
          \"num_unique_values\": 1,\n                \"samples\": [\n
\"Yes\"\n            ],\n           \"semantic_type\": \"\",\n
\"description\": \"\"\n             }\n      },\n      {\n         \"column\":
\"Apps\",\n           \"properties\": {\n               \"dtype\": \"number\",\n
\"std\": null,\n              \"min\": 3847,\n              \"max\": 3847,\n
\"num_unique_values\": 1,\n                \"samples\": [\n                3847\n
],\n           \"semantic_type\": \"\",\n               \"description\": \"\"\n
}\n      },\n      {\n         \"column\": \"Accept\",\n              \"properties\":
{\n          \"dtype\": \"number\",\n               \"std\": null,\n
\"min\": 3433,\n              \"max\": 3433,\n \"num_unique_values\": 1,\n
              \"samples\": [\n                3433\n           ],\n
\"semantic_type\": \"\",\n               \"description\": \"\"\n            }\
n      },\n      {\n         \"column\": \"Enroll\",\n             \"properties\":
{\n          \"dtype\": \"number\",\n               \"std\": null,\n
\"min\": 527,\n              \"max\": 527,\n    \"num_unique_values\":       1,\n
              \"samples\": [\n                527\n           ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n                            }\
n      },\n      {\n         \"column\": \"Top10perc\",\n
\"properties\": {\n              \"dtype\": \"number\",\n               \"std\":
```

```
\"number\",\n        \"std\": null,\n        \"min\": 14.3,\n
\"max\": 14.3,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          14.3\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"perc.alumni\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": null,\n        \"min\": 20,\n
\"max\": 20,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          20\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Expend\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": null,\n        \"min\": 7697,\n        \"max\": 7697,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          7697\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Grad.Rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": 118,\n        \"max\": 118,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          118\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```


```
sns.set_style('darkgrid')
g = sns.FacetGrid(df,hue="Private",palette='coolwarm',aspect=2)
g = g.map(plt.hist,'Grad.Rate',bins=20,alpha=0.7)
```




```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=2)

df.drop("Unnamed: 0",axis =1,inplace =True)
```

```
kmeans.fit(df.drop('Private',axis=1))
```

```
KMeans(n_clusters=2)
```

```
means=kmeans.cluster_centers_
print(means)
```

```
  6.59078947e+01]

  6.21935484e+01]]
```

```
[[1.99097222e+03 1.34700585e+03 5.01001462e+02 2.66637427e+01
  5.46023392e+01 2.19326316e+03 5.53080409e+02 1.06887091e+04
  4.37517398e+03 5.44059942e+02 1.26739474e+03 7.10745614e+01
  7.83391813e+01 1.38330409e+01 2.35716374e+01 9.58258772e+03

 [1.04349247e+04 6.95977419e+03 2.83176344e+03 3.41397849e+01
  6.45806452e+01 1.47810323e+04 3.07806452e+03 8.61637634e+03
  4.22773118e+03 5.88516129e+02 1.87936559e+03 8.43225806e+01
  8.97311828e+01 1.59774194e+01 1.66559140e+01 1.02307849e+04
```

```
def converter(cluster): if
     cluster=='Yes':
          return 1 else:
          return 0
df['Cluster'] = df['Private'].apply(converter) df.head(3)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 777,\n  \"fields\": [\ n     {\n
      \"column\": \"Private\",\n                                \"properties\": {\n
\"dtype\": \"category\",\n                \"num_unique_values\": 2,\n
\"samples\": [\n                \"No\",\n                \"Yes\"\n            ],\n
\"semantic_type\": \"\",\n                \"description\": \"\"\n            }\
n     },\n      {\n        \"column\": \"Apps\",\n             \"properties\": {\n
\"dtype\": \"number\",\n                \"std\": 3870,\n              \"min\": 81,\n
\"max\": 48094,\n              \"num_unique_values\": 711,\n
\"samples\": [\n                323,\n              1127\n            ],\n
\"semantic_type\": \"\",\n                \"description\": \"\"\n            }\
n     },\n      {\n        \"column\": \"Accept\",\n             \"properties\":
{\n          \"dtype\": \"number\",\n                \"std\": 2451,\n
\"min\": 72,\n    \"max\": 26330,\n                    \"num_unique_values\":
693,\n           \"samples\": [\n                689,\n              1623\
n          ],\n          \"semantic_type\": \"\",\n
\"description\": \"\"\n            }\n      },\n      {\n        \"column\":
\"Enroll\",\n          \"properties\": {\n              \"dtype\": \"number\",\n
\"std\": 929,\n              \"min\": 35,\n              \"max\": 6392,\n
\"num_unique_values\": 581,\n  \"samples\": [\n                                769,\n
223\n          ],\n          \"semantic_type\": \"\",\n
\"description\": \"\"\n            }\n      },\n      {\n        \"column\":
\"Top10perc\",\n           \"properties\": {\n              \"dtype\":
\"number\",\n             \"std\": 17,\n             \"min\": 1,\n
\"max\": 96,\n  \"num_unique_values\": 82,\n                    \"samples\": [\n
67,\n             23\n          ],\n          \"semantic_type\":
\"\",\n            \"description\": \"\"\n            }\n      },\n      {\n
\"column\": \"Top25perc\",\n              \"properties\": {\n
\"dtype\": \"number\",\n                \"std\": 19,\n              \"min\": 9,\n
\"max\": 100,\n              \"num_unique_values\": 89,\n
\"samples\": [\n                100,\n              65\n          ],\n
```

\"description\": \"\"\n }\n },\n {\n \"column\": \"perc.alumni\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 12,\n \"min\": 0,\n \"max\": 64,\n \"num_unique_values\": 61,\n \"samples\": [\n 12,\n 11\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Expend\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 5221,\n \"min\": 3186,\n \"max\": 56233,\n \"num_unique_values\": 744,\n \"samples\": [\n 17500,\n 13705\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Grad.Rate\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 17,\n \"min\": 10,\n \"max\": 118,\n \"num_unique_values\": 81,\n \"samples\": [\n 35,\n 60\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Cluster\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0,\n \"min\": 0,\n \"max\": 1,\n \"num_unique_values\": 2,\n \"samples\": [\n 0,\n 1\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.Private.value_counts()
```

```
Private Yes
         565
No       212
Name: count, dtype: int64
```

```
from sklearn.metrics import confusion_matrix,classification_report
print(confusion_matrix(df['Cluster'],kmeans.labels_))
print(classification_report(df['Cluster'],kmeans.labels_))
```

```
[[131  81]
 [553  12]]
                  precision        recall  f1-score        support

               0         0.19         0.62         0.29          212
               1         0.13         0.02         0.04          565

        accuracy                              0.18          777
       macro avg         0.16         0.32         0.16          777
    weighted avg         0.15         0.18         0.11          777
```

Conclusion:-

In this experiment, we understand to model evaluation using the Confusion Matrix for Logistic Regression. It helped them understand how to assess model accuracy, precision, recall, and overall performance in classification tasks.