

In [4]:

```
import pandas as pd
data = pd.read_csv('College.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Private         777 non-null    object
 1   Apps            777 non-null    int64
 2   Accept          777 non-null    int64
 3   Enroll          777 non-null    int64
 4   Top10perc       777 non-null    int64
 5   Top25perc       777 non-null    int64
 6   F.Undergrad     777 non-null    int64
 7   P.Undergrad     777 non-null    int64
 8   Outstate        777 non-null    int64
 9   Room.Board     777 non-null    int64
10   Books           777 non-null    int64
11   Personal        777 non-null    int64
12   PhD             777 non-null    int64
13   Terminal        777 non-null    int64
14   S.F.Ratio       777 non-null    float64
15   perc.alumni     777 non-null    int64
16   Expend          777 non-null    int64
17   Grad.Rate       777 non-null    int64
dtypes: float64(1), int64(16), object(1)
memory usage: 109.4+ KB
```

In [5]:

```
data.head()
```

Out[5]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Rc
0	Yes	1660	1232	721	23	52	2885	537	7440	
1	Yes	2186	1924	512	16	29	2683	1227	12280	
2	Yes	1428	1097	336	22	50	1036	99	11250	
3	Yes	417	349	137	60	89	510	63	12960	
4	Yes	193	146	55	16	44	249	869	7560	

In [7]:

```
x = data.drop(['Private'], axis=1)
y = data['Private']
```

In [9]:

```
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
scaled_x = scale.fit_transform(x)
```

In [10]:

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
encoded_y = encoder.fit_transform(data['Private'])
```

In [11]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(scaled_x, encoded_y)
```

In [12]:

```
from sklearn.svm import SVC
reg = SVC()
reg.fit(x_train, y_train)
y_pred = reg.predict(x_test)
```

In [14]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)
```

Out[14]:

```
0.9435897435897436
```

In [17]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_pred,y_test)
```

Out[17]:

```
array([[ 40,   5],
       [  6, 144]])
```

In [23]:

```
from sklearn.metrics import classification_report  
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.89	0.87	0.88	46
1	0.96	0.97	0.96	149
accuracy			0.94	195
macro avg	0.92	0.92	0.92	195
weighted avg	0.94	0.94	0.94	195

In [ ]: