

```

import numpy as np

import pandas as pd

import os

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn import svm

from sklearn.metrics import accuracy_score

from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

from sklearn.model_selection import cross_val_score

from sklearn import preprocessing

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

import joblib

from sklearn.metrics import accuracy_score

df = pd.read_csv(r"/content/collegePlace.csv")
df.head()

```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklog
0	22	Male	Electronics And Communication	1	8	1	
1	21	Female	Computer Science	0	7	1	
2	22	Female	Information Technology	1	6	0	

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 8 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Age                  2966 non-null  int64
1   Gender               2966 non-null  object
2   Stream               2966 non-null  object
3   Internships          2966 non-null  int64
4   CGPA                 2966 non-null  int64
5   Hostel               2966 non-null  int64
6   HistoryOfBacklogs    2966 non-null  int64
7   PlacedOrNot          2966 non-null  int64
dtypes: int64(6), object(2)
memory usage: 185.5+ KB

```

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

```

Age                0
Gender              0
Stream              0
Internships         0
CGPA                0
Hostel              0
HistoryOfBacklogs   0
PlacedOrNot         0
dtype: int64

```

```
def transformationplot(feature):

    plt.figure(figsize=(12,5))
    plt.subplot(1,2,1)
    sns.distplot(feature)

    transformationplot(np.log(df[ 'Age' ]))

df = df.replace(['Male'], [8])
df = df.replace(['Female'], [1])

df = df.replace(['Computer Science', 'Information Technology', 'Electronics And Communication', 'Mechanical', 'Electrical', 'Civil'], [0,1,2

df = df.drop(['Hostel'], axis=1)

df
```

	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot	
0	22	8	2	1	8	1	1	
1	21	1	0	0	7	1	1	
2	22	1	1	1	6	0	1	
3	21	8	1	0	8	1	1	
4	22	8	3	0	8	0	1	
...	
2961	23	8	1	0	7	0	0	
2962	23	8	3	1	7	0	0	
2963	22	8	1	1	7	0	0	
2964	22	8	0	1	7	0	0	
2965	23	8	5	0	8	0	1	

2966 rows × 7 columns

```
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['CGPA'], color='r')

plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['PlacedOrNot'], color='r')
```

```
<ipython-input-25-2aa230b900a8>:3: UserWarning:
```

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['CGPA'], color='r')
```

```
<ipython-input-25-2aa230b900a8>:7: UserWarning:
```

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

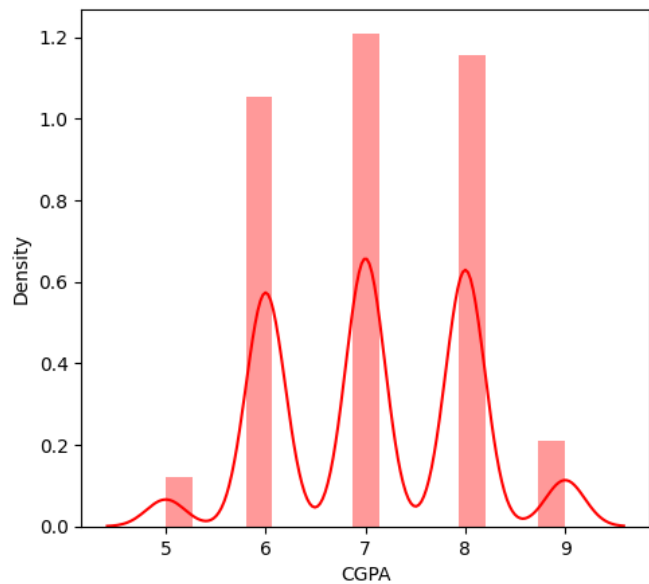
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

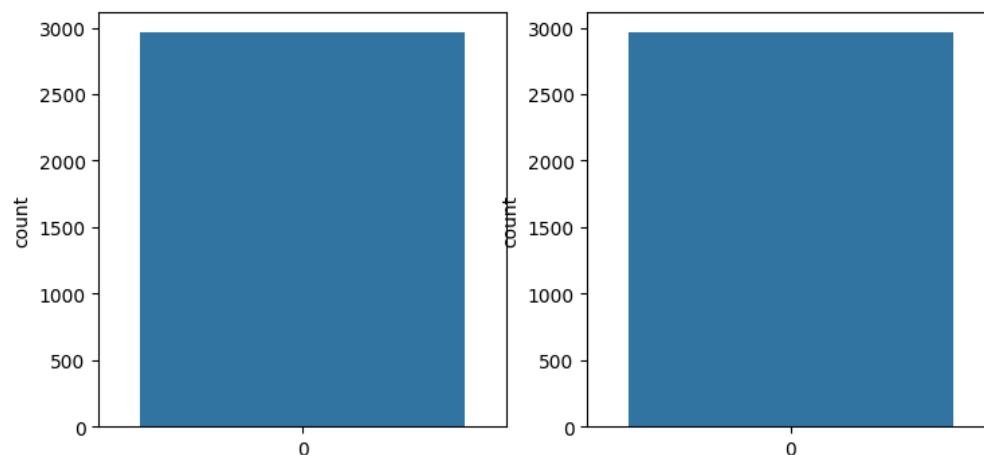
<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['PlacedOrNot'], color='r')
```

```
<Axes: xlabel='PlacedOrNot', ylabel='Density'>
```

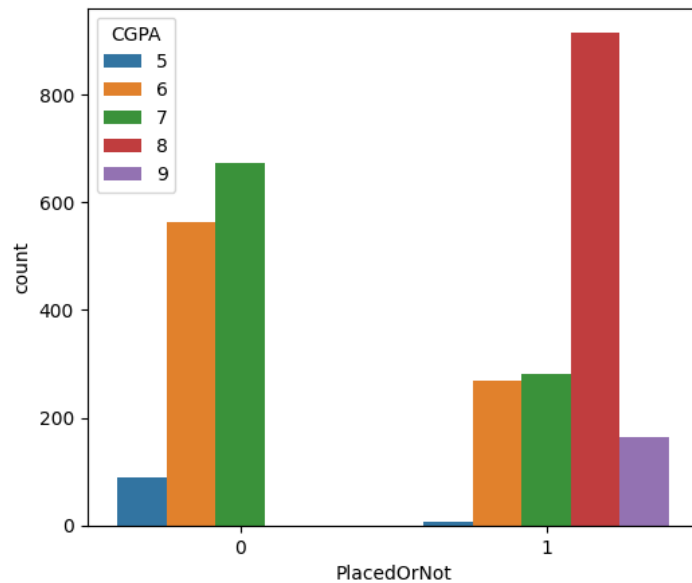


```
plt.figure(figsize=(18,4))
plt.subplot(1, 4, 1)
sns.countplot(df['Gender'])
plt.subplot(1, 4, 2)
sns.countplot(df['Stream'])
plt.show()
```



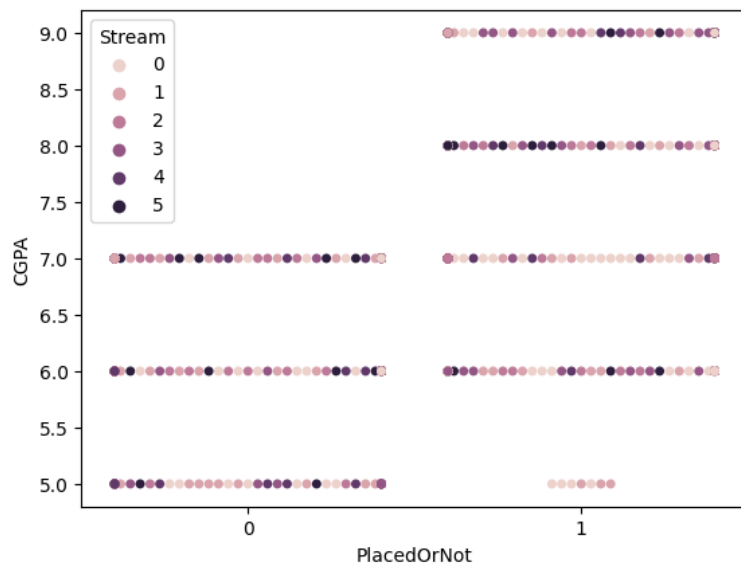
```
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(data=df, x="PlacedOrNot", hue="CGPA")
```

<Axes: xlabel='PlacedOrNot', ylabel='count'>



```
sns.swarmplot(x=df['PlacedOrNot'], y=df['CGPA'], hue=df['Stream'])
plt.show()
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 88.9% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 87.6% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 93.9% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 93.0% of the points cannot be placed; you may want to
warnings.warn(msg, UserWarning)
```



```
x = df.drop(['HistoryOfBacklogs'], axis=1)
y = df['Internships']
```

```
sc = StandardScaler()
```

```
x_bal = sc.fit_transform(x)
```

```
print(x_bal)
```

```
[[ 0.38813058  0.44540301  0.04008175  0.40044544  0.95719068  0.89979999]
 [-0.36675158 -2.24515772 -1.14874288 -0.95077319 -0.07631043  0.89979999]]
```

```
[ 0.38813058 -2.24515772 -0.55433057  0.40044544 -1.10981154  0.89979999]
...
[ 0.38813058  0.44540301 -0.55433057  0.40044544 -0.07631043 -1.11135809]
[ 0.38813058  0.44540301 -1.14874288  0.40044544 -0.07631043 -1.11135809]
[ 1.14301273  0.44540301  1.82331869 -0.95077319  0.95719068  0.89979999]]
```

```
names = x.columns
```

```
x_bal = pd.DataFrame(x_bal,columns=names)
```

```
print(x_bal)
```

	Age	Gender	Stream	Internships	CGPA	PlacedOrNot
0	0.388131	0.445403	0.040082	0.400445	0.957191	0.899800
1	-0.366752	-2.245158	-1.148743	-0.950773	-0.076310	0.899800
2	0.388131	-2.245158	-0.554331	0.400445	-1.109812	0.899800
3	-0.366752	0.445403	-0.554331	-0.950773	0.957191	0.899800
4	0.388131	0.445403	0.634494	-0.950773	0.957191	0.899800
...
2961	1.143013	0.445403	-0.554331	-0.950773	-0.076310	-1.111358
2962	1.143013	0.445403	0.634494	0.400445	-0.076310	-1.111358
2963	0.388131	0.445403	-0.554331	0.400445	-0.076310	-1.111358
2964	0.388131	0.445403	-1.148743	0.400445	-0.076310	-1.111358
2965	1.143013	0.445403	1.823319	-0.950773	0.957191	0.899800

```
[2966 rows x 6 columns]
```

```
print(df.columns)
```

```
Index(['Age', 'Gender', 'Stream', 'Internships', 'CGPA', 'HistoryOfBacklogs',
       'PlacedOrNot'],
      dtype='object')
```

```
if 'Gender' in df.columns and 'Stream' in df.columns:
```

```
    df = pd.get_dummies(df, columns=['Gender', 'Stream'], drop_first=True)
```

```
X = df.drop(['PlacedOrNot'], axis=1)
```

```
scaler = StandardScaler()
```

```
standardized_data = scaler.fit_transform(X)
```

```
X = standardized_data
```

```
Y = df['PlacedOrNot']
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

```
print("X_train shape:", X_train.shape)
```

```
print("Y_train shape:", Y_train.shape)
```

```
print("X_test shape:", X_test.shape)
```

```
print("Y_test shape:", Y_test.shape)
```

```
X_train shape: (2372, 10)
Y_train shape: (2372,)
X_test shape: (594, 10)
Y_test shape: (594,)
```

✓ 0s completed at 9:22 PM

● ×