dysi5mhay

July 28, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df=pd.read_csv("/content/13_placement.csv")
df
```

[2]:
```
     cgpa  placement_exam_marks  placed
0    7.19                  26.0       1
1    7.46                  38.0       1
2    7.54                  40.0       1
3    6.42                   8.0       1
4    7.23                  17.0       0
..    …                    …      …
995  8.87                  44.0       1
996  9.12                  65.0       1
997  4.89                  34.0       0
998  8.62                  46.0       1
999  4.90                  10.0       1

[1000 rows x 3 columns]
```

[3]:
```
df.head()
```

[3]:
```
   cgpa  placement_exam_marks  placed
0  7.19                  26.0       1
1  7.46                  38.0       1
2  7.54                  40.0       1
3  6.42                   8.0       1
4  7.23                  17.0       0
```

1 DATA CLEANING AND DATA PREPROCESSING

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   cgpa                 1000 non-null   float64
 1   placement_exam_marks 1000 non-null   float64
 2   placed               1000 non-null   int64
dtypes: float64(2), int64(1)
memory usage: 23.6 KB
```

[5]:
```
df.describe()
```

[5]:
```
              cgpa  placement_exam_marks       placed
count  1000.000000           1000.000000  1000.000000
mean      6.961240             32.225000     0.489000
std       0.615898             19.130822     0.500129
min       4.890000              0.000000     0.000000
25%       6.550000             17.000000     0.000000
50%       6.960000             28.000000     0.000000
75%       7.370000             44.000000     1.000000
max       9.120000            100.000000     1.000000
```

[6]:
```
df.columns
```

[6]:
```
Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```

[7]:
```
df1=df.dropna(axis=1)
df1
```

[7]:
```
     cgpa  placement_exam_marks  placed
0    7.19                  26.0       1
1    7.46                  38.0       1
2    7.54                  40.0       1
3    6.42                   8.0       1
4    7.23                  17.0       0
..    …                     …     …
995  8.87                  44.0       1
996  9.12                  65.0       1
997  4.89                  34.0       0
998  8.62                  46.0       1
999  4.90                  10.0       1

[1000 rows x 3 columns]
```

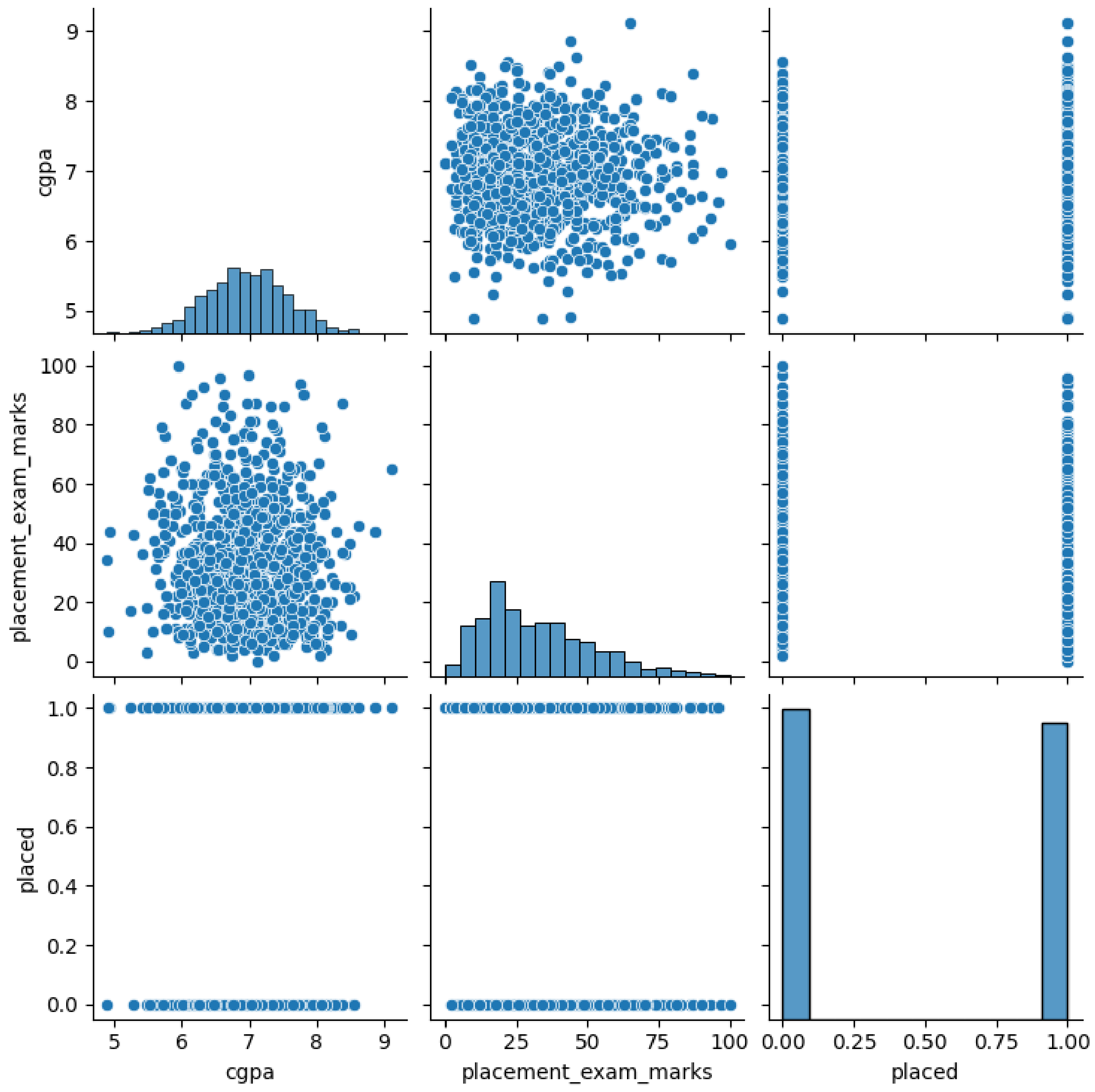
[8]:
```
df1.columns
```

```
[8]: Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```

2 EDA AND VISUALIZATION

```
[9]: sns.pairplot(df1)
```

```
[9]: <seaborn.axisgrid.PairGrid at 0x7bbed99e2020>
```

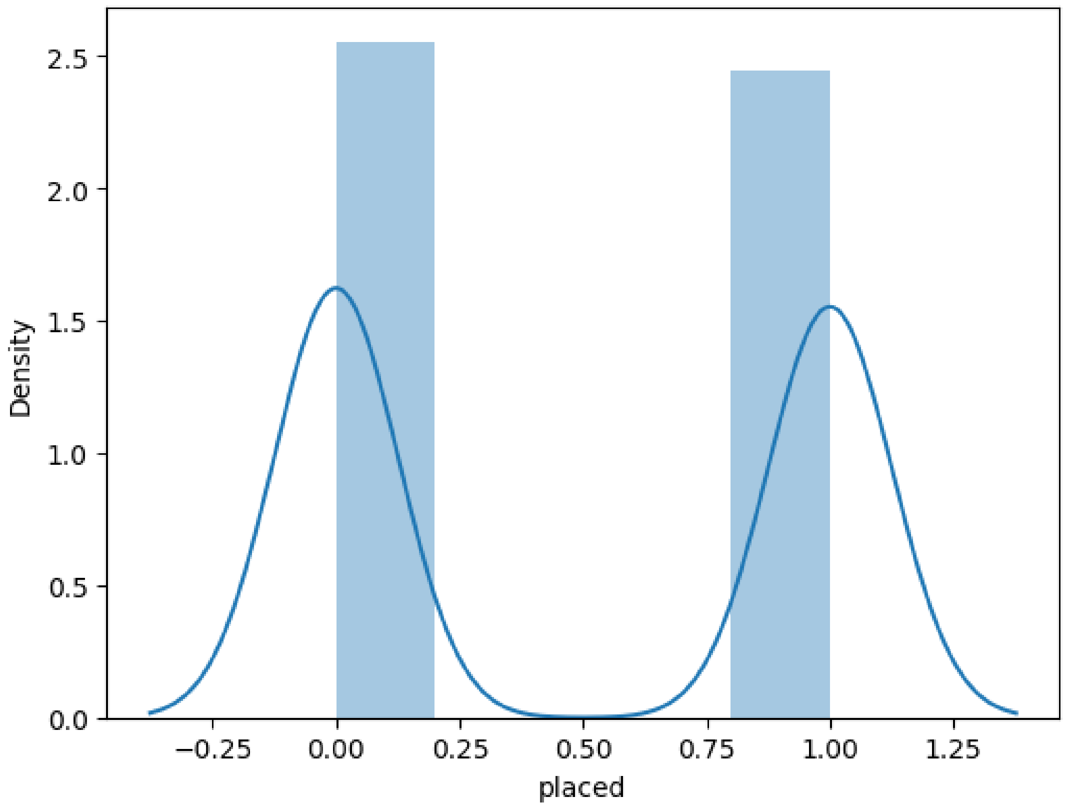


```
[10]: sns.distplot(df1['placed'])
```

```
<ipython-input-10-dc9f78aae914>:1: UserWarning:

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

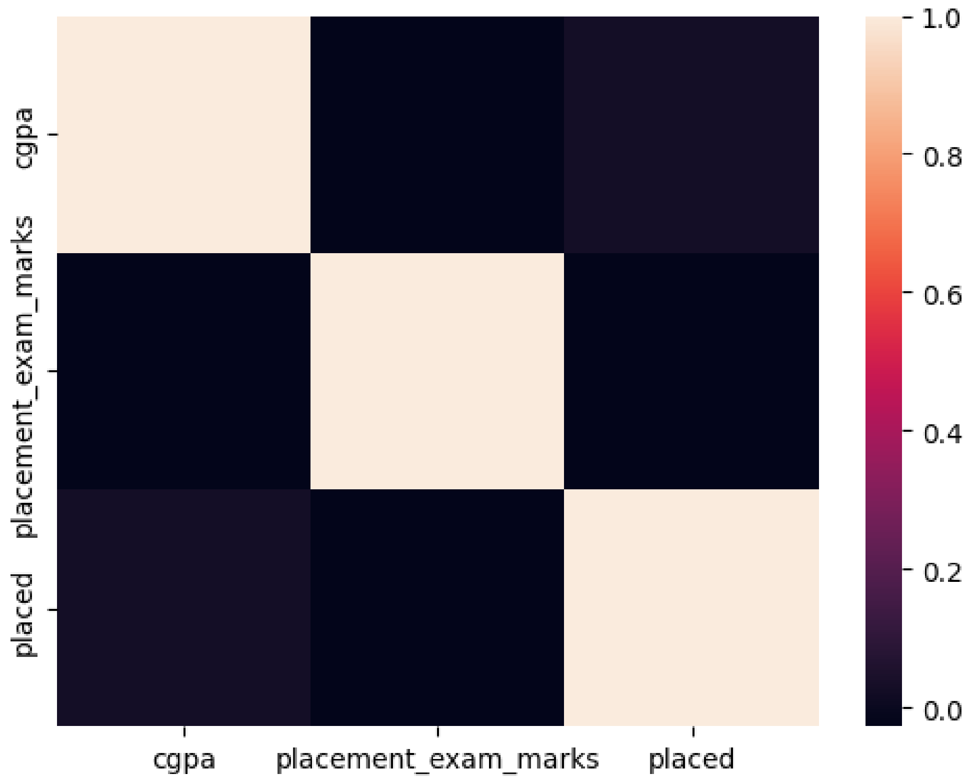
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1['placed'])
```

[10]:
```
<Axes: xlabel='placed', ylabel='Density'>
```



[11]:
```
sns.heatmap(df1.corr())
```

[11]:
```
<Axes: >
```



3 TO TRAIN THE MODEL AND MODEL BULDING

[12]:
```
x=df[['cgpa', 'placement_exam_marks']]
y=df['placed']
```

[13]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

[14]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[14]:
```
LinearRegression()
```

[15]:
```
lr.intercept_
```

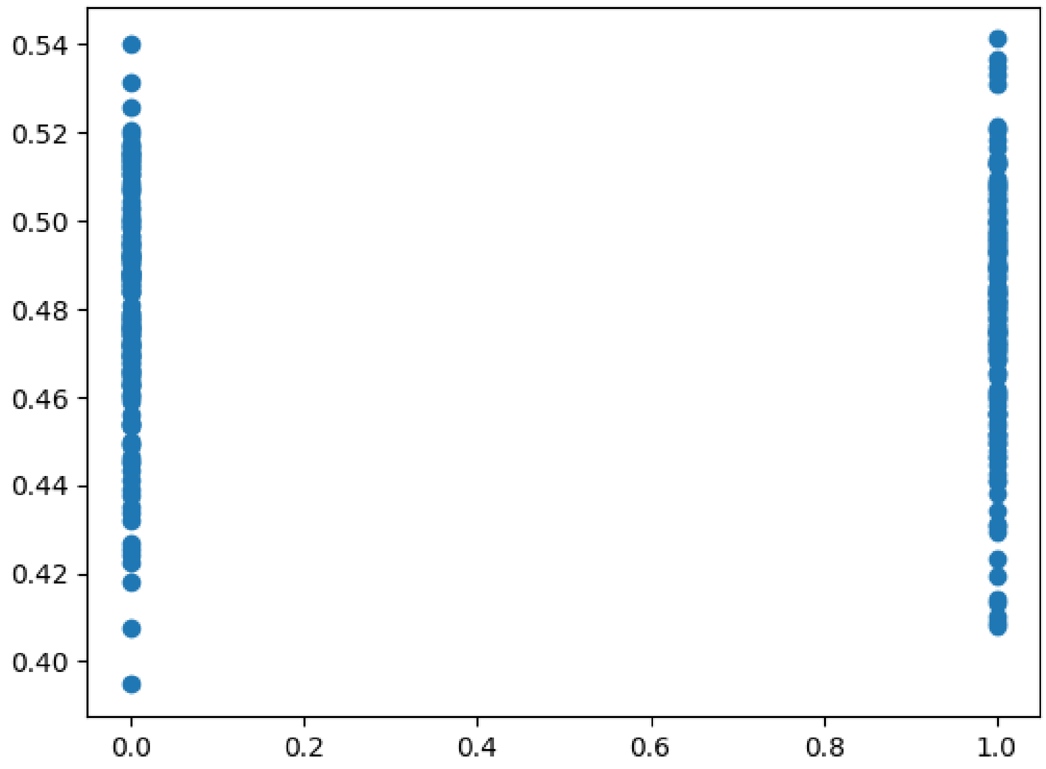
[15]:
```
0.2577539814877772
```

[16]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

```
[16]:                          Co-efficient
      cgpa                         0.035498
      placement_exam_marks        -0.000742
```

```
[17]: prediction =lr.predict(x_test)
      plt.scatter(y_test,prediction)
```

```
[17]: <matplotlib.collections.PathCollection at 0x7bbed2804a90>
```



4 ACCURACY

```
[18]: lr.score(x_test,y_test)
```

```
[18]: -0.00561591284914309
```

```
[19]: lr.score(x_train,y_train)
```

```
[19]: 0.002765450941136449
```

```
[20]: from sklearn.linear_model import Ridge,Lasso
      rr=Ridge(alpha=10)
```

```
rr.fit(x_train,y_train)
```

[20]:
```
Ridge(alpha=10)
```

[21]:
```
rr.score(x_train,y_train)
```

[21]:
```
0.0027629711257934897
```

[22]:
```
rr.score(x_test,y_test)
```

[22]:
```
-0.005435920006991557
```

[23]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

[23]:
```
Lasso(alpha=10)
```

[24]:
```
la.score(x_train,y_train)
```

[24]:
```
0.0
```

[25]:
```
la.score(x_test,y_test)
```

[25]:
```
-0.0025482988358924707
```