import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

# Input data files are available in the read-only "../input/" directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

df = pd.read\_csv(r'/content/collegePlace.csv')

df.head()

df.shape

df.info()

df.isna().sum()

df.Stream.unique()

plt.xticks(rotation = 90)

sns.barplot(x = df.Stream, y = df.PlacedOrNot)

df.Age.unique()

plt.figure(figsize = (12,7))

sns.barplot(x = df.Age, y = df.PlacedOrNot, hue = df.Gender)

plt.figure(figsize = (7,5))

sns.countplot(x = df.Age)

df.Age.value\_counts()

sns.barplot(x = df.Internships, y = df.PlacedOrNot)

df.Internships.value\_counts()

df.CGPA.value\_counts()

sns.barplot(x = df.CGPA, y = df.PlacedOrNot)

sns.barplot(x = df.Hostel, y = df.PlacedOrNot)

sns.barplot(x = df.Gender, y = df.PlacedOrNot)

sns.barplot(x = df.HistoryOfBacklogs, y = df.PlacedOrNot)

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df.Gender = le.fit\_transform(df.Gender)

df.Stream = le.fit\_transform(df.Stream)

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

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

y = df.PlacedOrNot

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import cross\_val\_score

cross\_val\_score(SVC(),x, y, cv = 3)

cross\_val\_score(DecisionTreeClassifier(), x, y, cv = 3)

cross\_val\_score(LogisticRegression(), x, y, cv = 3)

cross\_val\_score(RandomForestClassifier(n\_estimators=50), x, y, cv = 3)

cross\_val\_score(KNeighborsClassifier(),x, y ,cv = 3)

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2)

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot = True)