In [2]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn import svm
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [3]:
```python
dataset = pd.read_csv("Placement_Data_Full_Class.csv")
```

In [4]:
```python
dataset.head()
```

Out[4]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0

In [5]:
```python
# as salary and sl_no columns are not required for placement status prediction so we drop it
dataset.drop(['salary','sl_no'], axis=1, inplace=True)
```

In [6]:
```python
# missing values checking
dataset.isnull().sum()
```

Out[6]:
```
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
dtype: int64
```

In [7]:
```python
# checking column values data type
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   gender          215 non-null    object 
 1   ssc_p           215 non-null    float64
 2   ssc_b           215 non-null    object 
 3   hsc_p           215 non-null    float64
 4   hsc_b           215 non-null    object 
 5   hsc_s           215 non-null    object 
 6   degree_p        215 non-null    float64
 7   degree_t        215 non-null    object 
 8   workex          215 non-null    object 
 9   etest_p         215 non-null    float64
 10  specialisation  215 non-null    object 
 11  mba_p           215 non-null    float64
 12  status          215 non-null    object 
dtypes: float64(5), object(8)
memory usage: 22.0+ KB
```

Label Encoding Data

In [8]:
```python
# label encoding needs to be done to ensure all values in the dataset is numeric
# hsc_s, degree_t columns needs to be splitted into columns (get_dummies needs to be applied)
features_to_split = ['hsc_s','degree_t']
for feature in features_to_split:
    dummy = pd.get_dummies(dataset[feature])
    dataset = pd.concat([dataset, dummy], axis=1)
    dataset.drop(feature, axis=1, inplace=True)
```

In [9]:
```python
dataset
```

Out[9]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	degree_p	workex	etest_p	specialisation	mba_p	status	Arts	Commerce	Science	Comm&Mgmt	Others	Sci&Tech
0	M	67.00	Others	91.00	Others	58.00	No	55.0	Mkt&HR	58.80	Placed	0	1	0	0	0	1
1	M	79.33	Central	78.33	Others	77.48	Yes	86.5	Mkt&Fin	66.28	Placed	0	0	1	0	0	1
2	M	65.00	Central	68.00	Central	64.00	No	75.0	Mkt&Fin	57.80	Placed	1	0	0	1	0	0
3	M	56.00	Central	52.00	Central	52.00	No	66.0	Mkt&HR	59.43	Not Placed	0	0	1	0	0	1
4	M	85.80	Central	73.60	Central	73.30	No	96.8	Mkt&Fin	55.50	Placed	0	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	M	80.60	Others	82.00	Others	77.60	No	91.0	Mkt&Fin	74.49	Placed	0	1	0	1	0	0
211	M	58.00	Others	60.00	Others	72.00	No	74.0	Mkt&Fin	53.62	Placed	0	0	1	0	0	1
212	M	67.00	Others	67.00	Others	73.00	Yes	59.0	Mkt&Fin	69.72	Placed	0	1	0	1	0	0
213	F	74.00	Others	66.00	Others	58.00	No	70.0	Mkt&HR	60.23	Placed	0	1	0	1	0	0
214	M	62.00	Central	58.00	Others	53.00	No	89.0	Mkt&HR	60.22	Not Placed	0	0	1	1	0	0

215 rows × 17 columns

In [10]:
```python
dataset.rename(columns={"Others": "Other_Degree"},inplace=True)
```

In [11]:
```python
dataset
```

Out[11]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	degree_p	workex	etest_p	specialisation	mba_p	status	Arts	Commerce	Science	Comm&Mgmt	Other_Degree	Sci&Tech
0	M	67.00	Others	91.00	Others	58.00	No	55.0	Mkt&HR	58.80	Placed	0	1	0	0	0	1
1	M	79.33	Central	78.33	Others	77.48	Yes	86.5	Mkt&Fin	66.28	Placed	0	0	1	0	0	1
2	M	65.00	Central	68.00	Central	64.00	No	75.0	Mkt&Fin	57.80	Placed	1	0	0	1	0	0
3	M	56.00	Central	52.00	Central	52.00	No	66.0	Mkt&HR	59.43	Not Placed	0	0	1	0	0	1
4	M	85.80	Central	73.60	Central	73.30	No	96.8	Mkt&Fin	55.50	Placed	0	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	M	80.60	Others	82.00	Others	77.60	No	91.0	Mkt&Fin	74.49	Placed	0	1	0	1	0	0
211	M	58.00	Others	60.00	Others	72.00	No	74.0	Mkt&Fin	53.62	Placed	0	0	1	0	0	1
212	M	67.00	Others	67.00	Others	73.00	Yes	59.0	Mkt&Fin	69.72	Placed	0	1	0	1	0	0
213	F	74.00	Others	66.00	Others	58.00	No	70.0	Mkt&HR	60.23	Placed	0	1	0	1	0	0
214	M	62.00	Central	58.00	Others	53.00	No	89.0	Mkt&HR	60.22	Not Placed	0	0	1	1	0	0

215 rows × 17 columns

In [12]:
```python
encoder = LabelEncoder() # to encode string to the values like 0,1,2 etc.
```

In [13]:
```python
columns_to_encode = ['gender','ssc_b', 'hsc_b','workex','specialisation','status']
for column in columns_to_encode:
    dataset[column] = encoder.fit_transform(dataset[column])
```

In [14]:
```python
dataset
```

Out[14]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	degree_p	workex	etest_p	specialisation	mba_p	status	Arts	Commerce	Science	Comm&Mgmt	Other_Degree	Sci&Tech
0	1	67.00	1	91.00	1	58.00	0	55.0	1	58.80	1	0	1	0	0	0	1
1	1	79.33	0	78.33	1	77.48	1	86.5	0	66.28	1	0	0	1	0	0	1
2	1	65.00	0	68.00	0	64.00	0	75.0	0	57.80	1	1	0	0	1	0	0
3	1	56.00	0	52.00	0	52.00	0	66.0	1	59.43	0	0	0	1	0	0	1
4	1	85.80	0	73.60	0	73.30	0	96.8	0	55.50	1	0	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	1	80.60	1	82.00	1	77.60	0	91.0	0	74.49	1	0	1	0	1	0	0
211	1	58.00	1	60.00	1	72.00	0	74.0	0	53.62	1	0	0	1	0	0	1
212	1	67.00	1	67.00	1	73.00	1	59.0	0	69.72	1	0	1	0	1	0	0
213	0	74.00	1	66.00	1	58.00	0	70.0	1	60.23	1	0	1	0	1	0	0
214	1	62.00	0	58.00	1	53.00	0	89.0	1	60.22	0	0	0	1	1	0	0

215 rows × 17 columns

In [15]:
```python
dataset.describe()
```

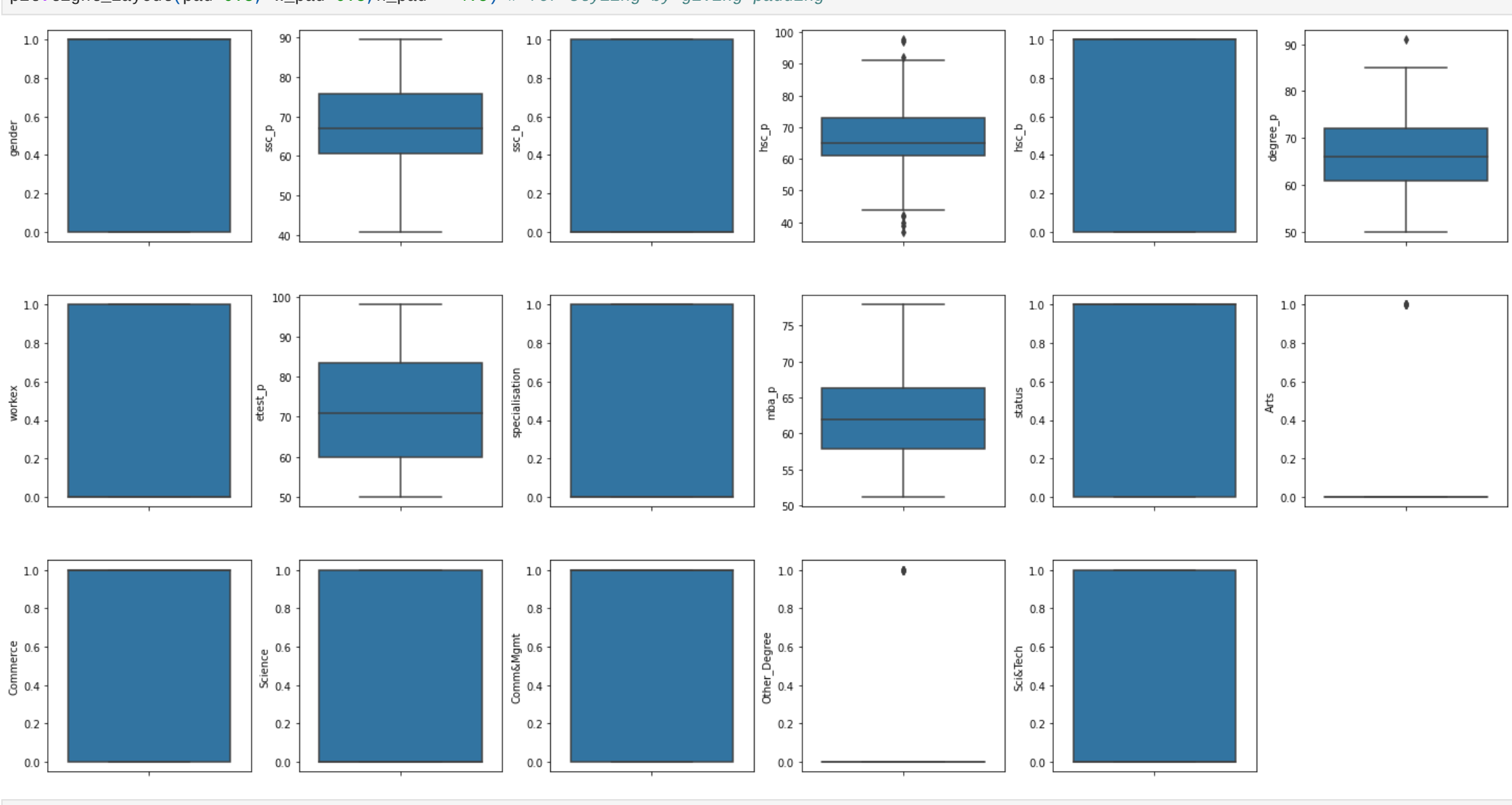
Out[15]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	degree_p	workex	etest_p	specialisation	mba_p	status	Arts	Commerce	Science	Com
count	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	21
mean	0.646512	67.303395	0.460465	66.333163	0.609302	66.370186	0.344186	72.100558	0.441860	62.278186	0.688372	0.051163	0.525581	0.423256	
std	0.479168	10.827205	0.499598	10.897509	0.489045	7.358743	0.476211	13.275956	0.497767	5.833385	0.464240	0.220844	0.500510	0.495228	
min	0.000000	40.890000	0.000000	37.000000	0.000000	50.000000	0.000000	50.000000	0.000000	51.210000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	60.600000	0.000000	60.900000	0.000000	61.000000	0.000000	60.000000	0.000000	57.945000	0.000000	0.000000	0.000000	0.000000	
50%	1.000000	67.000000	0.000000	65.000000	1.000000	66.000000	0.000000	71.000000	0.000000	62.000000	1.000000	0.000000	1.000000	0.000000	
75%	1.000000	75.700000	1.000000	73.000000	1.000000	72.000000	1.000000	83.500000	1.000000	66.255000	1.000000	0.000000	1.000000	1.000000	
max	1.000000	89.400000	1.000000	97.700000	1.000000	91.000000	1.000000	98.000000	1.000000	77.890000	1.000000	1.000000	1.000000	1.000000	

Checking for Outliers

In [16]:
```python
fig, axs = plt.subplots(ncols=6,nrows=3,figsize=(20,10))
index = 0
axs = axs.flatten()
for k,v in dataset.items():
    sns.boxplot(y=v, ax=axs[index])
    index+=1

fig.delaxes(axs[index])
plt.tight_layout(pad=0.3, w_pad=0.5,h_pad = 4.5) # for styling by giving padding
```



In [17]:
```python
# deleting some outliers in 2 columns degree_p and hsc_p
dataset = dataset[~(dataset['degree_p']>=90)]
dataset = dataset[~(dataset['hsc_p']>=95)]
```

Checking for Correlation

In [18]:
```python
dataset.corr()
```

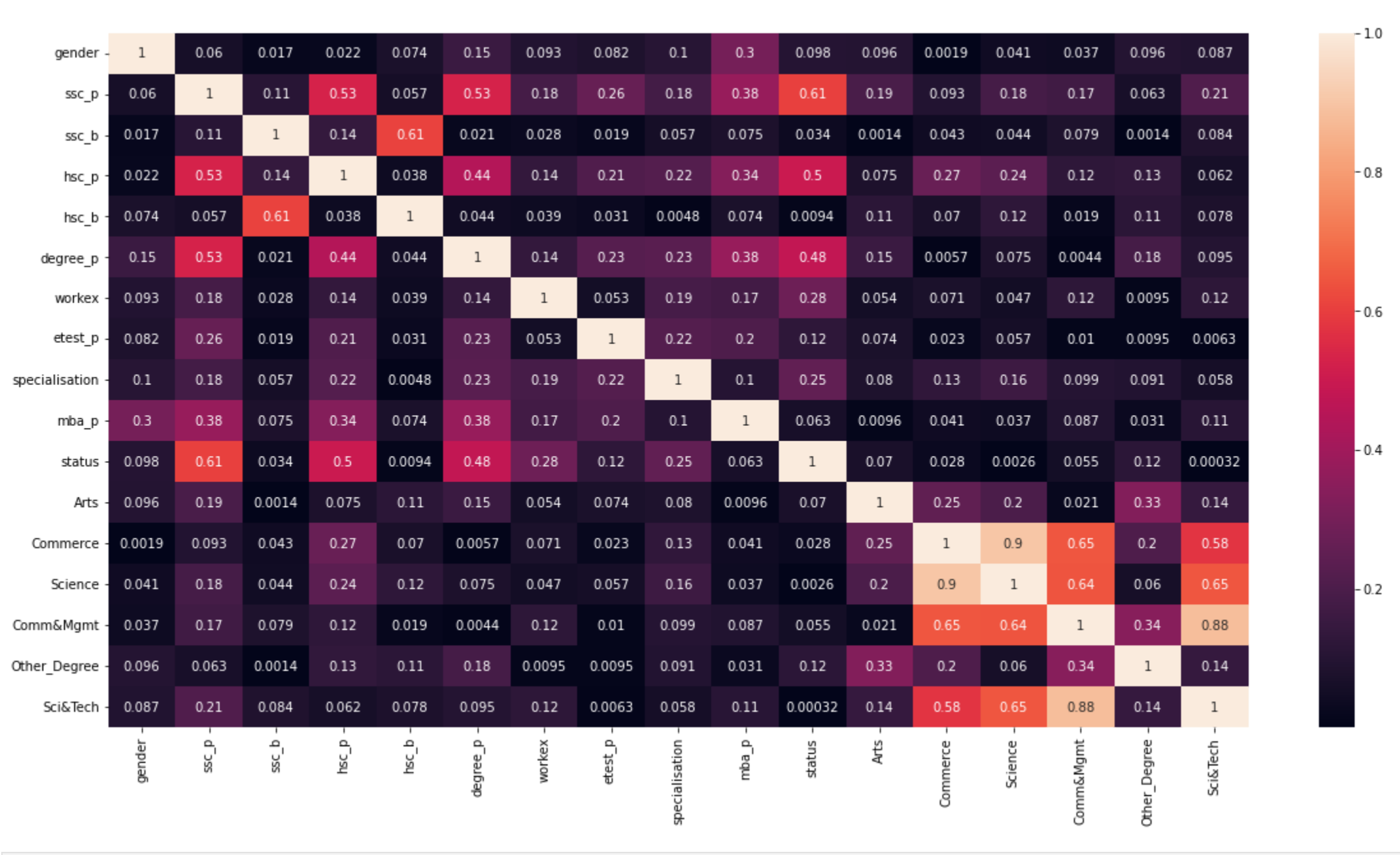
Out[18]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	degree_p	workex	etest_p	specialisation	mba_p	status	Arts	Commerce	Science	Comm&Mgmt
gender	1.000000	-0.059818	0.017052	-0.022187	0.074438	-0.154679	0.093325	0.081765	-0.103355	-0.298466	0.098189	-0.096386	0.001870	0.041426	-0.036801
ssc_p	-0.059818	1.000000	0.107995	0.528111	0.056672	0.528753	0.183073	0.264009	-0.177436	0.377438	0.605381	-0.194514	-0.093283	0.181772	-0.168282
ssc_b	0.017052	0.107995	1.000000	-0.140332	0.608493	0.020828	-0.027916	-0.018739	-0.057356	0.074653	0.033717	-0.001410	-0.042586	0.043708	-0.078842
hsc_p	-0.022187	0.528111	-0.140332	1.000000	-0.038259	0.443595	0.135144	0.208809	-0.222405	0.335610	0.499777	-0.074931	0.267073	-0.236466	0.121441
hsc_b	0.074438	0.056672	0.608493	-0.038259	1.000000	0.043618	0.039061	0.031316	0.004762	0.073936	0.009393	-0.114855	-0.069985	0.122407	-0.019492
degree_p	-0.154679	0.528753	0.020828	0.443595	0.043618	1.000000	0.135100	0.226353	-0.232618	0.376261	0.479557	-0.153777	-0.005676	0.074850	-0.004369
workex	0.093325	0.183073	-0.027916	0.135144	0.039061	0.135100	1.000000	0.052862	-0.187200	0.174951	0.279091	0.054259	-0.070916	0.047346	-0.118781
etest_p	0.081765	0.264009	-0.018739	0.208809	0.031316	0.226353	0.052862	1.000000	-0.222765	0.203663	0.122770	-0.073539	-0.023192	0.056508	-0.010486
specialisation	-0.103355	-0.177436	-0.057356	-0.222405	0.004762	-0.232618	-0.187200	-0.222765	1.000000	-0.100456	-0.251043	-0.080368	-0.126684	0.164258	-0.098640
mba_p	-0.298466	0.377438	0.074653	0.335610	0.073936	0.376261	0.174951	0.203663	-0.100456	1.000000	0.063197	0.009640	-0.040704	0.036839	-0.087109
status	0.098189	0.605381	0.033717	0.499777	0.009393	0.479557	0.279091	0.122770	-0.251043	0.063197	1.000000	-0.069693	0.028377	0.002618	0.054545
Arts	-0.096386	-0.194514	-0.001410	-0.074931	-0.114855	-0.153777	0.054259	-0.073539	-0.080368	0.009640	-0.069693	1.000000	-0.247575	-0.198994	-0.021492
Commerce	0.001870	-0.093283	-0.042586	0.267073	-0.069985	-0.005676	-0.070916	-0.023192	-0.126684	-0.040704	0.028377	-0.247575	1.000000	-0.900226	0.646272
Science	0.041426	0.181772	0.043708	-0.236466	0.122407	0.074850	0.047346	0.056508	0.164258	0.036839	0.002618	-0.198994	-0.900226	1.000000	-0.644039
Comm&Mgmt	-0.036801	-0.168282	-0.078842	0.121441	-0.019492	-0.004369	-0.118781	-0.010486	-0.098640	-0.087109	0.054545	-0.021492	0.646272	-0.644039	1.000000
Other_Degree	-0.096386	-0.063459	-0.001410	-0.132137	-0.114855	-0.180476	0.009501	0.009482	0.090868	-0.031441	-0.115435	0.328810	-0.204973	0.059556	-0.340428
Sci&Tech	0.086960	0.208907	0.083707	-0.061747	0.077977	0.094883	0.120296	0.006296	0.058387	0.107435	0.000324	-0.141863	-0.577836	0.648230	-0.882467

In [19]:
```python
# heatmap for checking correlation or linearity

plt.figure(figsize=(20,10))
sns.heatmap(dataset.corr().abs(), annot=True)
```

Out[19]:
```
<AxesSubplot:>
```



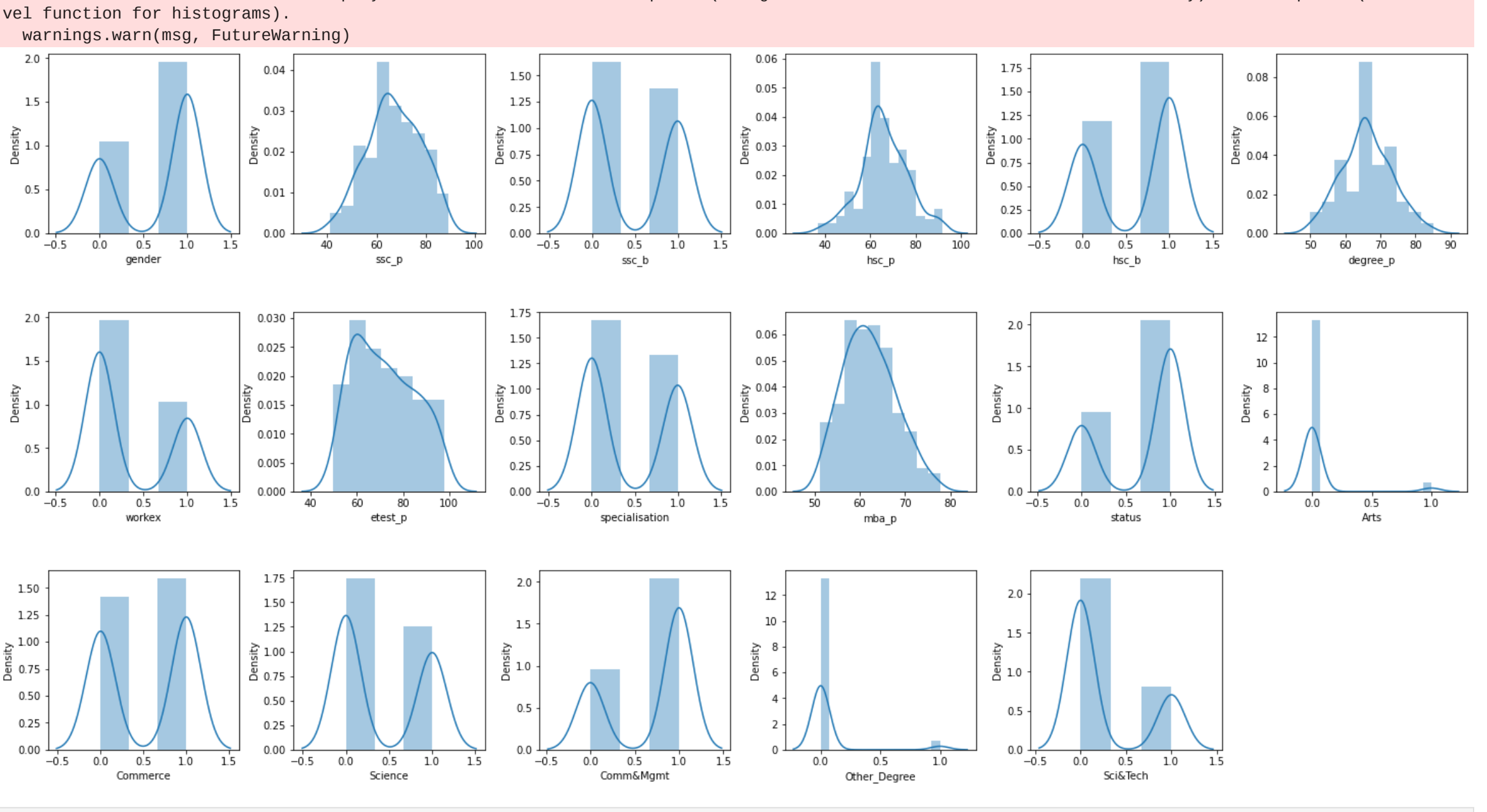
In [20]:
```python
dataset.shape
```

Out[20]:
```
(212, 17)
```

In [21]:
```python
# checking distributions of all features
fig, axs = plt.subplots(ncols=6,nrows=3,figsize=(20,10))
index = 0
axs = axs.flatten()
for k,v in dataset.items():
    sns.distplot(v, ax=axs[index])
    index+=1

fig.delaxes(axs[index]) # deleting the 18th figure
plt.tight_layout(pad=0.3, w_pad=0.2,h_pad = 4.5)
```

```
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Green zone\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [22]:
```python
x = dataset.loc[:,dataset.columns!='status'] # all features are used
y = dataset.loc[:, 'status'] # label is status of placement
```

In [23]:
```python
x
```

Out[23]:

	gender	ssc_p	ssc_b	hsc_p	hsc_b	degree_p	workex	etest_p	specialisation	mba_p	Arts	Commerce	Science	Comm&Mgmt	Other_Degree	Sci&Tech
0	1	67.00	1	91.00	1	58.00	0	55.0	1	58.80	0	1	0	0	0	1
1	1	79.33	0	78.33	1	77.48	1	86.5	0	66.28	0	0	1	0	0	1
2	1	65.00	0	68.00	0	64.00	0	75.0	0	57.80	1	0	0	1	0	0
3	1	56.00	0	52.00	0	52.00	0	66.0	1	59.43	0	0	1	0	0	1
4	1	85.80	0	73.60	0	73.30	0	96.8	0	55.50	0	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	1	80.60	1	82.00	1	77.60	0	91.0	0	74.49	0	1	0	1	0	0
211	1	58.00	1	60.00	1	72.00	0	74.0	0	53.62	0	0	1	0	0	1
212	1	67.00	1	67.00	1	73.00	1	59.0	0	69.72	0	1	0	1	0	0
213	0	74.00	1	66.00	1	58.00	0	70.0	1	60.23	0	1	0	1	0	0
214	1	62.00	0	58.00	1	53.00	0	89.0	1	60.22	0	0	1	1	0	0

212 rows × 16 columns

In [24]:
```python
y
```

Out[24]:
```
0      1
1      1
2      1
3      0
4      1
      ..
210    1
211    1
212    1
213    1
214    0
Name: status, Length: 212, dtype: int32
```

In [25]:
```python
sc= StandardScaler()
x_scaled = sc.fit_transform(x) # for standardising the features
x_scaled = pd.DataFrame(x_scaled)
```

In [26]:
```python
x_train,x_test, y_train, y_test = train_test_split(x_scaled,y,test_size=0.18, random_state=0)
```

Using Logistic Regression

In [27]:
```python
lr = LogisticRegression()
```

In [28]:
```python
lr.fit(x_train, y_train)
```

Out[28]:
```
LogisticRegression()
```

In [29]:
```python
y_pred = lr.predict(x_test)
```

In [30]:
```python
y_pred
```

Out[30]:
```
array([1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0,
       1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0])
```

In [31]:
```python
y_test
```

Out[31]:
```
209    1
38     1
90     1
192    1
150    1
76     1
97     0
138    1
5      0
84     1
56     1
144    0
159    0
113    1
75     0
203    1
127    1
12     0
169    0
157    1
167    0
201    0
211    1
189    0
184    0
18     0
214    0
15     1
87     0
72     1
7      1
64     1
142    1
98     1
137    1
161    0
34     0
153    1
91     0
Name: status, dtype: int32
```

In [32]:
```python
accuracy_score(y_test, y_pred)
```

Out[32]:
```
0.8717948717948718
```

In [33]:
```python
lr.score(x_train,y_train)
```

Out[33]:
```
0.9132947976878613
```

In [34]:
```python
confusion_matrix(y_test, y_pred)
```

Out[34]:
```
array([[14,  3],
       [ 2, 20]], dtype=int64)
```

In [35]:
```python
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.82      0.85        17
           1       0.87      0.91      0.89        22

    accuracy                           0.87        39
   macro avg       0.87      0.87      0.87        39
weighted avg       0.87      0.87      0.87        39
```

Using Naive Bayes Classifier - Gaussian Naive Bayes

In [36]:
```python
nbclassifier = GaussianNB()
```

In [37]:
```python
nbclassifier.fit(x_train, y_train)
```

Out[37]:
```
GaussianNB()
```

In [38]:
```python
y_pred_nb = nbclassifier.predict(x_test)
```

In [39]:
```python
accuracy_score(y_test, y_pred_nb)
```

Out[39]:
```
0.8461538461538461
```

In [40]:
```python
nbclassifier.score(x_train, y_train)
```

Out[40]:
```
0.8554913294797688
```

In [41]:
```python
confusion_matrix(y_test, y_pred_nb)
```

Out[41]:
```
array([[13,  4],
       [ 2, 20]], dtype=int64)
```

Using SVM Linear Kernel

In [42]:
```python
clf = svm.SVC(kernel="linear")
```

In [43]:
```python
clf.fit(x_train, y_train)
```

Out[43]:
```
SVC(kernel='linear')
```

In [44]:
```python
y_pred_svm = clf.predict(x_test)
```

In [45]:
```python
accuracy_score(y_test, y_pred_svm)
```

Out[45]:
```
0.8974358974358975
```

In [46]:
```python
clf.score(x_train, y_train)
```

Out[46]:
```
0.9017341040462428
```

In [47]:
```python
confusion_matrix(y_test, y_pred_svm)
```

Out[47]:
```
array([[15,  2],
       [ 2, 20]], dtype=int64)
```

In [48]:
```python
print(classification_report(y_test, y_pred_svm))
```

```
              precision    recall  f1-score   support

           0       0.88      0.88      0.88        17
           1       0.91      0.91      0.91        22

    accuracy                           0.90        39
   macro avg       0.90      0.90      0.90        39
weighted avg       0.90      0.90      0.90        39
```

So, Naive Bayes was better for not overfitting the data

SVM gave better accuracy with least difference in score.

So, Our final model would use SVM for Student Placement Prediction.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: