problem-1-1

March 23, 2023

```
[3]: import pandas as pd
     import seaborn as sns
     import numpy as np
     import matplotlib.pyplot as plt
     import matplotlib as mpl
     %matplotlib inline
     mpl.style.use('ggplot')
```

```
[4]: df=pd.read_csv(r"C:\Users\Ranveer\Downloads\Placement_Data_Full_Class.csv")
```

```
[5]: df.shape
```

```
[5]: (215, 15)
```

```
[4]: df.sample(10)
```

```
[4]:      sl_no gender  ssc_p    ssc_b  hsc_p    hsc_b     hsc_s  degree_p  \
     146    147      M   62.0  Central   63.0   Others   Science      66.0
     78      79      M   84.0   Others   90.9   Others   Science      64.5
     95      96      M   73.0  Central   78.0   Others  Commerce      65.0
     10      11      M   58.0  Central   61.0  Central  Commerce      60.0
     154    155      M   53.0  Central   63.0   Others   Science      60.0
     2        3      M   65.0  Central   68.0  Central      Arts      64.0
     96      97      F   76.0  Central   70.0  Central   Science      76.0
     48      49      M   63.0   Others   62.0   Others  Commerce      68.0
     130    131      M   62.0  Central   65.0   Others  Commerce      60.0
     63      64      M   61.0   Others   70.0   Others  Commerce      64.0

           degree_t workex  etest_p specialisation  mba_p      status    salary
     146  Comm&Mgmt     No    85.00         Mkt&HR  55.14      Placed  233000.0
     78    Sci&Tech     No    86.04        Mkt&Fin  59.42      Placed  270000.0
     95   Comm&Mgmt    Yes    95.46        Mkt&Fin  62.16      Placed  420000.0
     10   Comm&Mgmt    Yes    62.00         Mkt&HR  60.85      Placed  260000.0
     154  Comm&Mgmt    Yes    70.00        Mkt&Fin  53.20      Placed  250000.0
     2    Comm&Mgmt     No    75.00        Mkt&Fin  57.80      Placed  250000.0
     96   Comm&Mgmt    Yes    66.00        Mkt&Fin  64.44      Placed  300000.0
     48   Comm&Mgmt     No    64.00        Mkt&Fin  62.46      Placed  250000.0
```

```
130  Comm&Mgmt      No     84.00         Mkt&Fin   64.15  Not Placed        NaN
63   Comm&Mgmt      No     68.50          Mkt&HR   59.50  Not Placed        NaN
```

[5]:

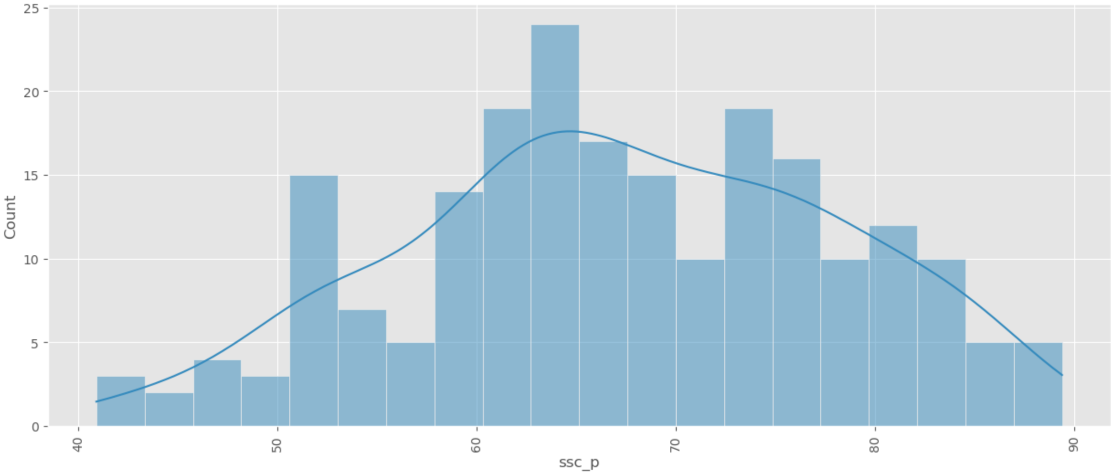
```
df['ssc_p'].value_counts().shape
```

[5]:

```
(103,)
```

[6]:

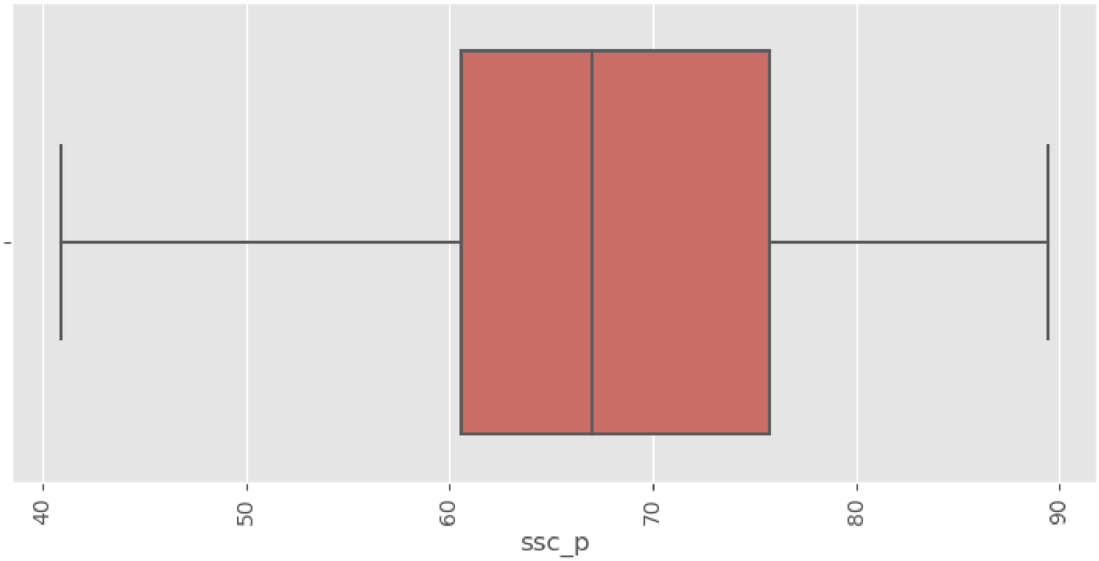
```
plt.figure(figsize=(15,6))
sns.histplot(df['ssc_p'], kde = True,bins=20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



[7]:

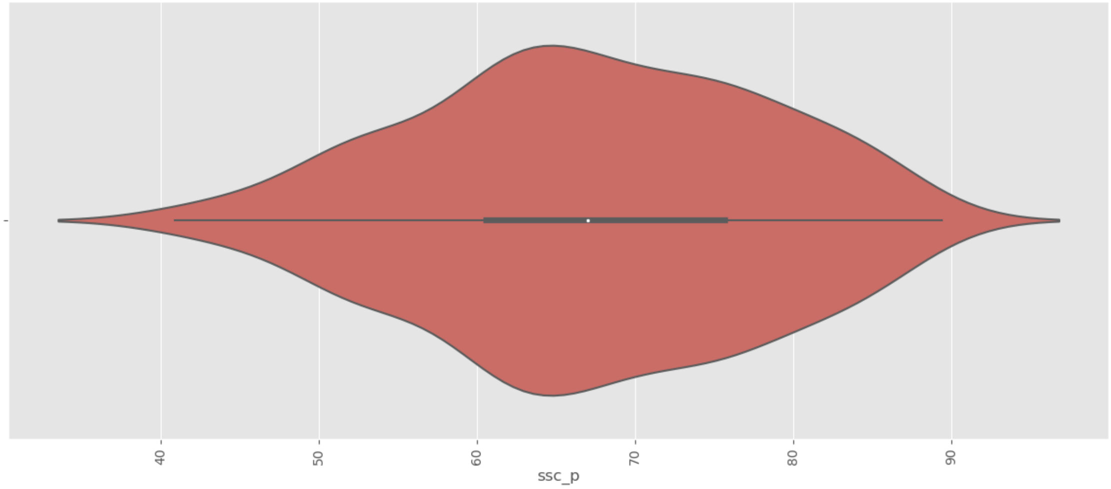
```
plt.figure(figsize=(9,4))
sns.boxplot(df['ssc_p'], data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

```
C:\Users\Ranveer\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



[8]:
```
plt.figure(figsize=(15,6))
sns.violinplot(df['ssc_p'], data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

```
C:\Users\Ranveer\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



[9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sl_no           215 non-null    int64
 1   gender          215 non-null    object
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object
 6   hsc_s           215 non-null    object
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object
 9   workex          215 non-null    object
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object
 14  salary          148 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

[10]:

```
df['ssc_b'].value_counts().shape
```

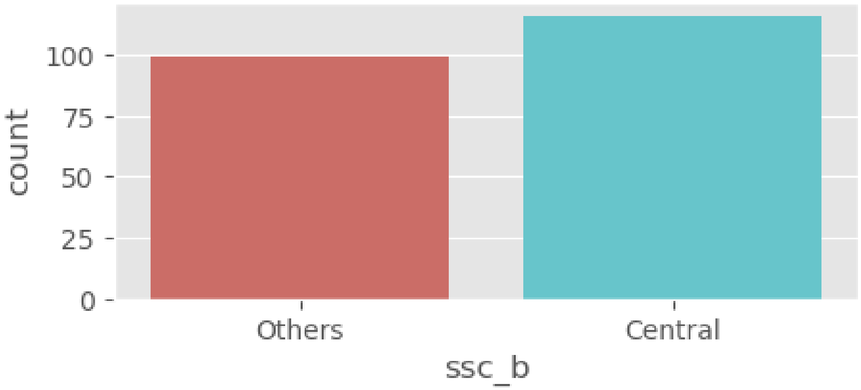
[10]:

```
(2,)
```

[11]:

```
plt.figure(figsize=(5,2))
sns.countplot(df['ssc_b'], data = df, palette = 'hls')
plt.show()
```

```
C:\Users\Ranveer\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



```
[12]: from sklearn.preprocessing import LabelEncoder
      encoder = LabelEncoder()
```

```
[13]: df['ssc_b'] = encoder.fit_transform(df['ssc_b'])
```

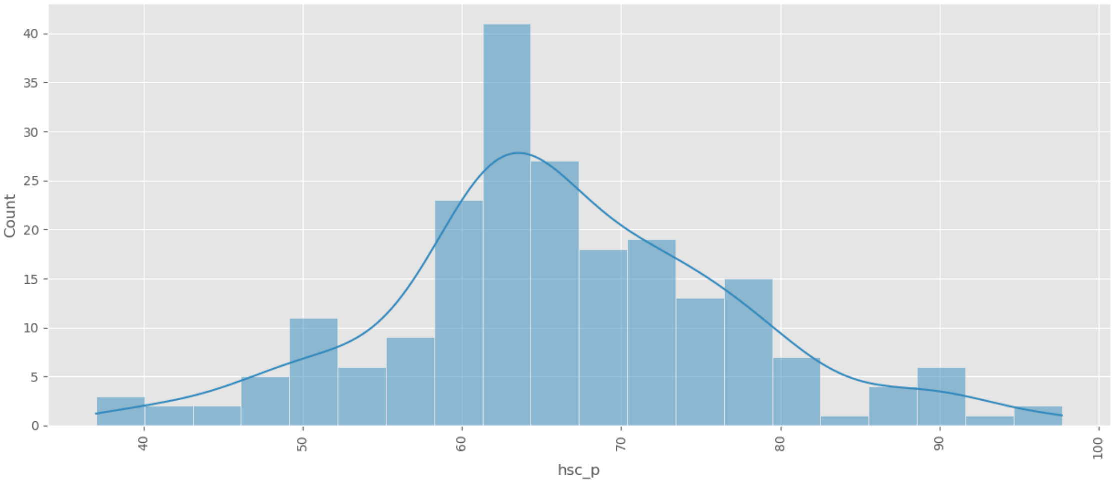
```
[14]: df.sample(5)
```

```
[14]:      sl_no gender  ssc_p  ssc_b  hsc_p    hsc_b     hsc_s  degree_p  \
      155    156      M  51.57      1  74.66   Others  Commerce     59.90
      23      24      F  77.40      1  60.00   Others   Science     64.74
      160    161      M  87.00      0  74.00  Central   Science     65.00
      83      84      M  84.00      1  79.00   Others   Science     68.00
      44      45      F  77.00      1  73.00   Others  Commerce     81.00

            degree_t workex  etest_p specialisation  mba_p      status    salary
      155  Comm&Mgmt    Yes    56.15          Mkt&HR  65.99  Not Placed       NaN
      23    Sci&Tech    Yes    92.00         Mkt&Fin  63.62      Placed  300000.0
      160   Sci&Tech    Yes    75.00          Mkt&HR  72.29      Placed  300000.0
      83    Sci&Tech    Yes    84.00         Mkt&Fin  66.69      Placed  300000.0
      44   Comm&Mgmt    Yes    89.00         Mkt&Fin  69.70      Placed  200000.0
```

```
[15]: df['hsc_p'].value_counts().shape
```

```
[15]: (97,)
```

```
[16]: plt.figure(figsize=(15,6))
      sns.histplot(df['hsc_p'], kde = True, bins = 20, palette = 'hls')
      plt.xticks(rotation = 90)
      plt.show()
```



[17]:
```
df['hsc_b'].value_counts().shape
```

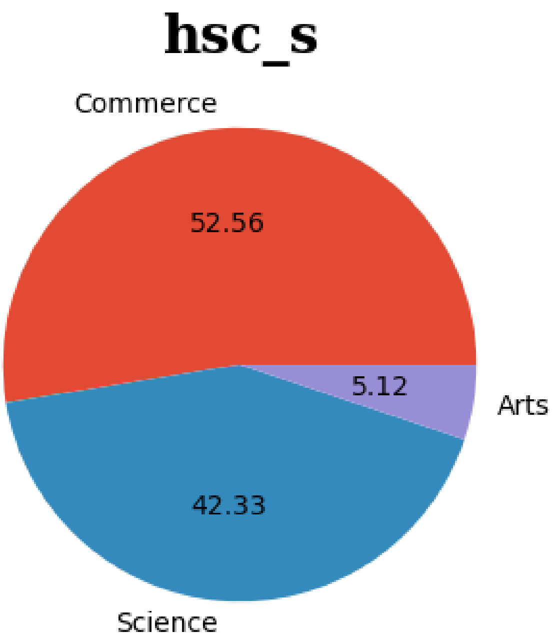
[17]:
```
(2,)
```

[18]:
```
import plotly.graph_objects as go

fig = go.Figure(go.Bar(
            x=df['hsc_b'].value_counts(),
            y=['Central', 'Others'],
            orientation='h'))
fig.update_layout(
autosize=False,
width=500,
height=250)
fig.show()
```

[19]:
```
df['hsc_s'].value_counts().shape
```

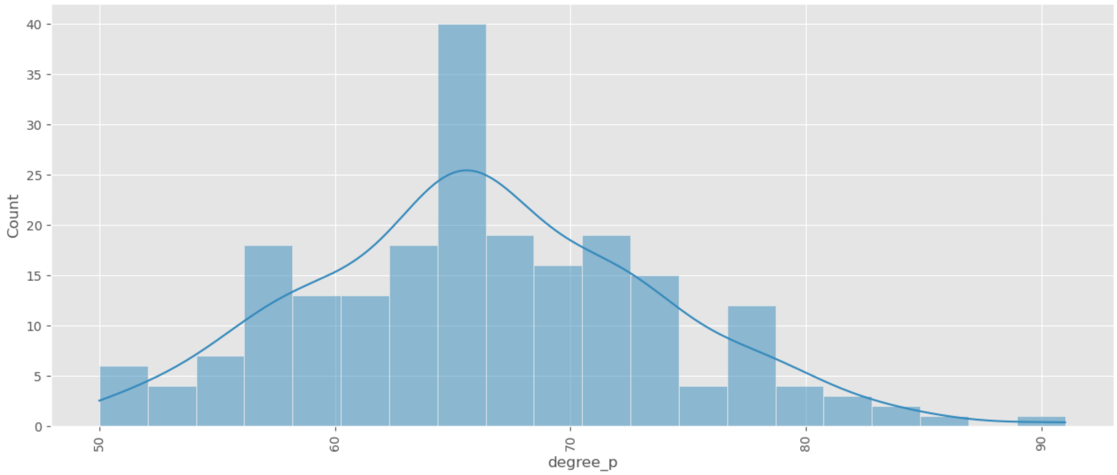
[19]:
```
(3,)
```

[20]:
```
plt.figure(figsize=(5,4))
plt.pie(df['hsc_s'].value_counts(), labels=df['hsc_s'].value_counts().
 ↪index,autopct='%.2f')
hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('hsc_s', size=20, **hfont)
plt.show()
```



[21]:

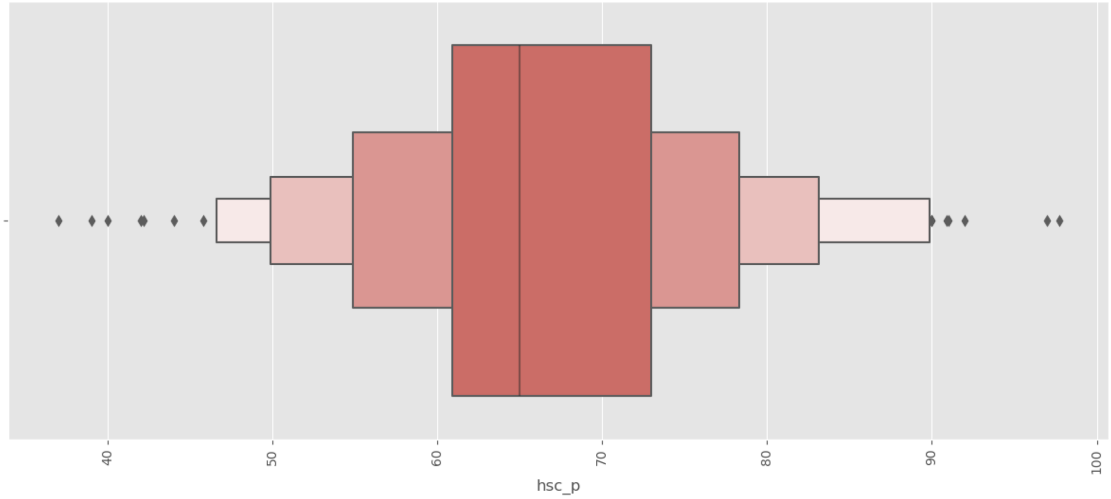
```
plt.figure(figsize=(15,6))
sns.histplot(df['degree_p'], kde = True,bins=20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



[11]:

```
plt.figure(figsize=(15,6))
sns.boxenplot(df['hsc_p'], data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

```
C:\Users\Ranveer\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

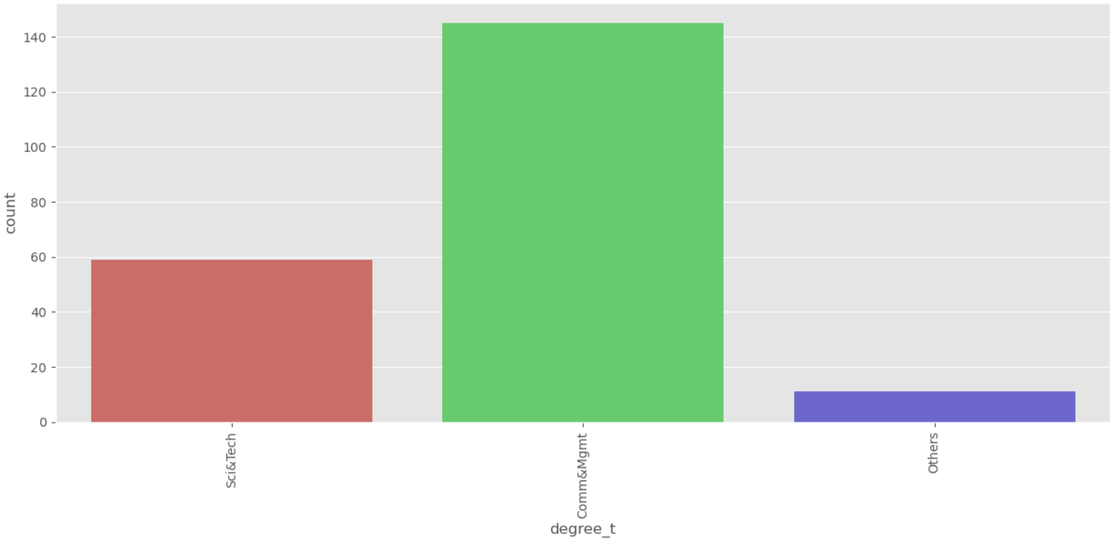


```
[22]: df['degree_t'].value_counts().shape
```

```
[22]: (3,)
```

```
[23]: plt.figure(figsize=(15,6))
      sns.countplot(df['degree_t'], data = df, palette = 'hls')
      plt.xticks(rotation = 90)
      plt.show()
```

```
C:\Users\Ranveer\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



[24]:
```
df['workex'].value_counts().shape
```

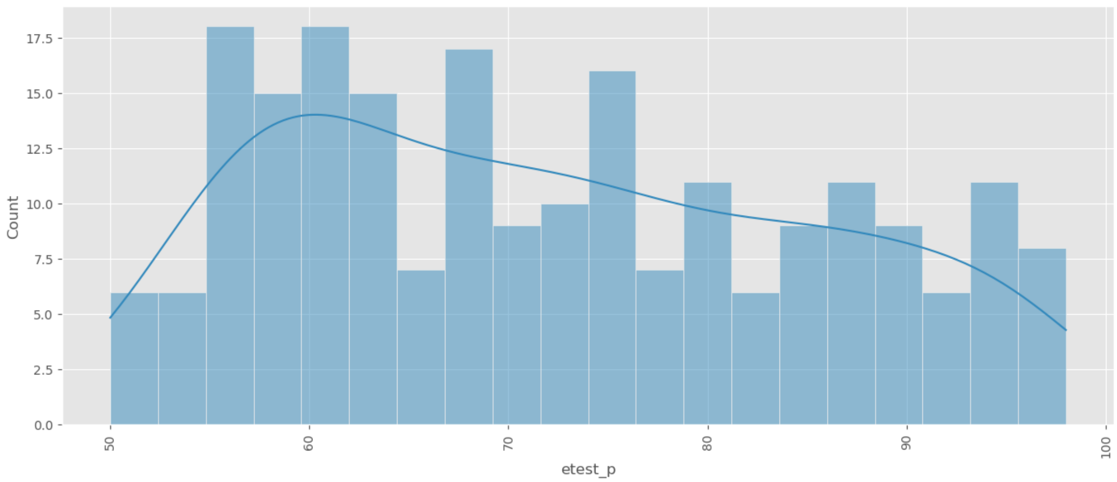
[24]:
```
(2,)
```

[25]:
```
import plotly.graph_objects as go

fig = go.Figure(go.Bar(
            x=df['workex'].value_counts(),
            y=['Yes', 'No'],
            orientation='h'))
fig.update_layout(
autosize=False,
width=500,
height=250)
fig.show()
```

[26]:
```
df['etest_p'].value_counts().shape
```

[26]:
```
(100,)
```

[27]:
```
plt.figure(figsize=(15,6))
sns.histplot(df['etest_p'], kde = True,bins=20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



[28]:
```
df['specialisation'].value_counts().shape
```

[28]:
```
(2,)
```

[29]:
```
import plotly.graph_objects as go

fig = go.Figure(go.Bar(
            x=df['specialisation'].value_counts(),
            y=['Mkt&Fin', 'Mkt&HR'],
            orientation='h'))
fig.update_layout(
autosize=False,
width=500,
height=250)
fig.show()
```

[30]:
```
df['specialisation'] = encoder.fit_transform(df['specialisation'])
```

[31]:
```
df.sample(10)
```

[31]:
```
     sl_no gender  ssc_p  ssc_b  hsc_p    hsc_b    hsc_s  degree_p  \
107    108      M  82.00      1  90.00   Others Commerce     83.00
36      37      M  51.00      0  44.00  Central Commerce     57.00
185    186      F  88.00      0  72.00  Central  Science     78.00
60      61      M  74.00      0  70.00  Central  Science     72.00
169    170      M  59.96      1  42.16   Others  Science     61.26
213    214      F  74.00      1  66.00   Others Commerce     58.00
44      45      F  77.00      1  73.00   Others Commerce     81.00
206    207      M  41.00      0  42.00  Central  Science     60.00
32      33      F  61.00      0  81.00  Central Commerce     66.40
```

```
195    196       M  66.00       0  76.00  Central  Commerce     72.00

      degree_t workex  etest_p  specialisation  mba_p      status    salary
107  Comm&Mgmt     No    80.00               1  73.52      Placed  200000.0
36   Comm&Mgmt     No    64.00               0  51.45  Not Placed       NaN
185     Others     No    82.00               1  71.43      Placed  252000.0
60   Comm&Mgmt    Yes    60.00               0  57.24      Placed  260000.0
169   Sci&Tech     No    54.48               1  65.48  Not Placed       NaN
213  Comm&Mgmt     No    70.00               1  60.23      Placed  204000.0
44   Comm&Mgmt    Yes    89.00               0  69.70      Placed  200000.0
206  Comm&Mgmt     No    97.00               0  53.39  Not Placed       NaN
32   Comm&Mgmt     No    50.89               1  62.21      Placed  278000.0
195  Comm&Mgmt    Yes    84.00               1  58.95      Placed  275000.0
```

[32]:
```python
df['status'].value_counts().shape
```

[32]:
```
(2,)
```

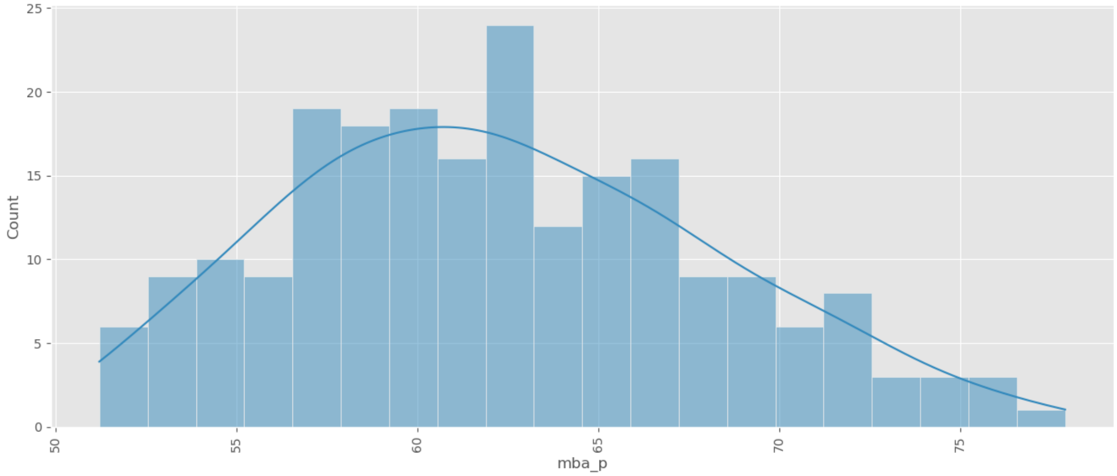
[33]:
```python
df['status'] = encoder.fit_transform(df['status'])
```

[34]:
```python
import plotly.graph_objects as go

fig = go.Figure(go.Bar(
            x=df['status'].value_counts(),
            y=['Not Placed', 'Placed'],
            orientation='h'))
fig.update_layout(
autosize=False,
width=500,
height=250)
fig.show()
```

[35]:
```python
df['mba_p'].value_counts().shape
```

[35]:
```
(205,)
```

[36]:
```python
plt.figure(figsize=(15,6))
sns.histplot(df['mba_p'], kde = True,bins=20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



[37]:

```
df.sample(20)
```

[37]:

```
     sl_no gender  ssc_p  ssc_b  hsc_p    hsc_b     hsc_s  degree_p  \
114    115      M   65.0      0  68.00   Others   Science     69.00
213    214      F   74.0      1  66.00   Others  Commerce     58.00
41      42      F   74.0      1  63.16   Others  Commerce     65.00
29      30      M   62.0      0  67.00  Central  Commerce     58.00
150    151      M   71.0      0  58.66  Central   Science     58.00
204    205      F   74.0      1  73.00   Others  Commerce     73.00
144    145      M   52.0      1  50.00   Others      Arts     61.00
146    147      M   62.0      0  63.00   Others   Science     66.00
86      87      M   62.0      1  63.00   Others  Commerce     64.00
44      45      F   77.0      1  73.00   Others  Commerce     81.00
122    123      F   66.5      0  66.80  Central      Arts     69.30
19      20      M   60.0      1  67.00   Others      Arts     70.00
39      40      M   81.0      1  68.00   Others   Science     64.00
12      13      F   47.0      0  55.00   Others   Science     65.00
3        4      M   56.0      0  52.00  Central   Science     52.00
159    160      M   52.0      0  49.00   Others  Commerce     58.00
43      44      M   87.0      1  87.00   Others  Commerce     68.00
98      99      F   69.0      0  73.00  Central  Commerce     65.00
26      27      M   71.0      1  79.00   Others  Commerce     66.00
128    129      M   80.4      0  73.40  Central   Science     77.72

      degree_t workex  etest_p  specialisation  mba_p  status    salary
114  Comm&Mgmt     No    53.70               1  55.01       1  250000.0
213  Comm&Mgmt     No    70.00               1  60.23       1  204000.0
41   Comm&Mgmt    Yes    65.00               1  69.76       0       NaN
29   Comm&Mgmt     No    77.00               0  51.29       0       NaN
150   Sci&Tech    Yes    56.00               0  61.30       1  690000.0
```

```
204  Comm&Mgmt    Yes    80.00               0  67.69       1  210000.0
144  Comm&Mgmt     No    60.00               0  58.52       0       NaN
146  Comm&Mgmt     No    85.00               1  55.14       1  233000.0
86   Comm&Mgmt     No    67.00               0  57.03       1  220000.0
44   Comm&Mgmt    Yes    89.00               0  69.70       1  200000.0
122  Comm&Mgmt    Yes    80.40               0  71.00       1  236000.0
19   Comm&Mgmt    Yes    50.48               0  77.89       1  236000.0
39    Sci&Tech     No    93.00               0  62.56       1  411000.0
12   Comm&Mgmt     No    62.00               1  65.04       0       NaN
3     Sci&Tech     No    66.00               1  59.43       0       NaN
159  Comm&Mgmt     No    62.00               1  60.59       0       NaN
43   Comm&Mgmt     No    95.00               1  62.90       1  300000.0
98   Comm&Mgmt     No    70.00               0  57.31       1  220000.0
26   Comm&Mgmt    Yes    94.00               0  57.55       1  240000.0
128   Sci&Tech    Yes    81.20               1  76.26       1  400000.0
```

[38]:
```
df['hsc_b'] = encoder.fit_transform(df['hsc_b'])
```

[39]:
```
df['workex'] = encoder.fit_transform(df['workex'])
```

[40]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

[41]:
```
df['gender'] = encoder.fit_transform(df['gender'])
```

[42]:
```
df['hsc_s'] = encoder.fit_transform(df['hsc_s'])
```

[43]:
```
df['degree_t'] = encoder.fit_transform(df['degree_t'])
```

[44]:
```
df.sample(10)
```

[44]:
```
     sl_no  gender  ssc_p  ssc_b  hsc_p  hsc_b  hsc_s  degree_p  degree_t  \
76      77       0  66.50      1  70.40      0      0     71.93         0
143    144       1  77.67      1  64.89      1      1     70.67         0
17      18       0  55.00      0  67.00      0      1     64.00         0
140    141       1  65.00      0  64.80      1      1     69.50         0
160    161       1  87.00      0  74.00      0      2     65.00         2
166    167       1  62.00      1  62.00      1      1     60.00         0
8        9       1  73.00      0  79.00      0      1     72.00         0
180    181       1  65.00      0  71.50      1      1     62.80         0
23      24       0  77.40      1  60.00      1      2     64.74         2
91      92       1  52.00      0  57.00      0      1     50.80         0

     workex  etest_p  specialisation  mba_p  status    salary
76        0    61.00               0  64.27       1  230000.0
143       0    89.00               0  60.39       1  300000.0
17        0    60.00               0  67.28       0       NaN
```

```
140       1    56.00              0   56.94        1   265000.0
160       1    75.00              1   72.29        1   300000.0
166       1    63.00              1   52.38        1   240000.0
8         0    91.34              0   61.29        1   231000.0
180       1    57.00              0   56.60        1   265000.0
23        1    92.00              0   63.62        1   300000.0
91        0    67.00              1   62.79        0        NaN
```

[45]:
```
df.describe()
```

[45]:
```
            sl_no      gender       ssc_p       ssc_b       hsc_p       hsc_b  \
count  215.000000  215.000000  215.000000  215.000000  215.000000  215.000000
mean   108.000000    0.646512   67.303395    0.460465   66.333163    0.609302
std     62.209324    0.479168   10.827205    0.499598   10.897509    0.489045
min      1.000000    0.000000   40.890000    0.000000   37.000000    0.000000
25%     54.500000    0.000000   60.600000    0.000000   60.900000    0.000000
50%    108.000000    1.000000   67.000000    0.000000   65.000000    1.000000
75%    161.500000    1.000000   75.700000    1.000000   73.000000    1.000000
max    215.000000    1.000000   89.400000    1.000000   97.700000    1.000000

            hsc_s    degree_p    degree_t      workex     etest_p  \
count  215.000000  215.000000  215.000000  215.000000  215.000000
mean     1.372093   66.370186    0.600000    0.344186   72.100558
std      0.580978    7.358743    0.890238    0.476211   13.275956
min      0.000000   50.000000    0.000000    0.000000   50.000000
25%      1.000000   61.000000    0.000000    0.000000   60.000000
50%      1.000000   66.000000    0.000000    0.000000   71.000000
75%      2.000000   72.000000    2.000000    1.000000   83.500000
max      2.000000   91.000000    2.000000    1.000000   98.000000

       specialisation       mba_p      status         salary
count      215.000000  215.000000  215.000000     148.000000
mean         0.441860   62.278186    0.688372  288655.405405
std          0.497767    5.833385    0.464240   93457.452420
min          0.000000   51.210000    0.000000  200000.000000
25%          0.000000   57.945000    0.000000  240000.000000
50%          0.000000   62.000000    1.000000  265000.000000
75%          1.000000   66.255000    1.000000  300000.000000
max          1.000000   77.890000    1.000000  940000.000000
```

[46]:
```
df.corr()
```

[46]:
```
                   sl_no    gender     ssc_p     ssc_b     hsc_p     hsc_b  \
sl_no           1.000000  0.074306 -0.078155  0.027214 -0.085711  0.116887
gender          0.074306  1.000000 -0.068969  0.019429 -0.021334  0.065945
ssc_p          -0.078155 -0.068969  1.000000  0.116194  0.511472  0.066996
ssc_b           0.027214  0.019429  0.116194  1.000000 -0.137013  0.605883
```

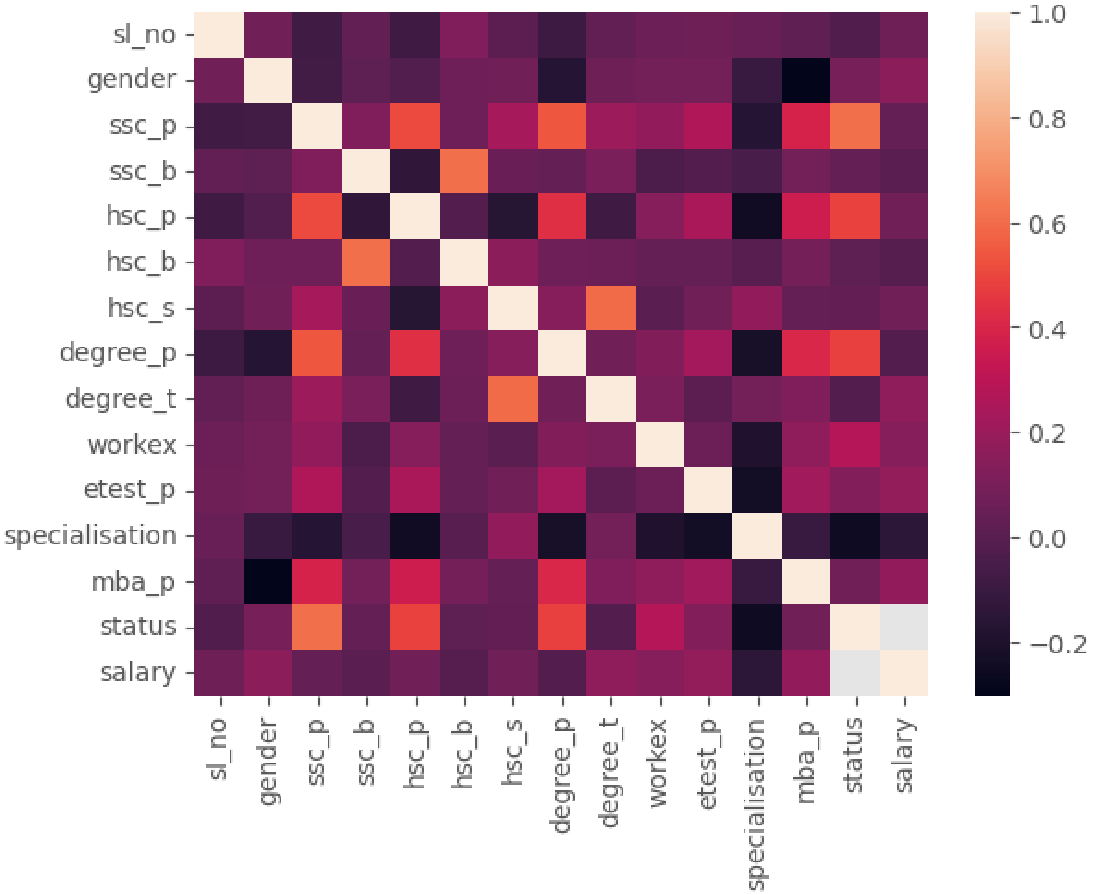
```
hsc_p          -0.085711 -0.021334  0.511472 -0.137013  1.000000 -0.019548
hsc_b           0.116887  0.065945  0.066996  0.605883 -0.019548  1.000000
hsc_s           0.009955  0.071827  0.236364  0.050919 -0.164091  0.152227
degree_p       -0.088281 -0.173217  0.538404  0.038070  0.434206  0.067229
degree_t        0.025651  0.061345  0.205896  0.100863 -0.086450  0.057960
workex          0.059151  0.085153  0.175675 -0.040744  0.141025  0.038357
etest_p         0.063636  0.084294  0.261993 -0.018991  0.245113  0.039108
specialisation  0.046630 -0.106160 -0.172536 -0.051565 -0.241630  0.002232
mba_p           0.022327 -0.300531  0.388478  0.083120  0.354823  0.090201
status         -0.026859  0.090670  0.607889  0.037297  0.491228  0.016945
salary          0.063764  0.158912  0.035330  0.005539  0.076819 -0.007549

                   hsc_s  degree_p  degree_t    workex   etest_p  \
sl_no           0.009955 -0.088281  0.025651  0.059151  0.063636
gender          0.071827 -0.173217  0.061345  0.085153  0.084294
ssc_p           0.236364  0.538404  0.205896  0.175675  0.261993
ssc_b           0.050919  0.038070  0.100863 -0.040744 -0.018991
hsc_p          -0.164091  0.434206 -0.086450  0.141025  0.245113
hsc_b           0.152227  0.067229  0.057960  0.038357  0.039108
hsc_s           1.000000  0.137276  0.596300  0.007856  0.075643
degree_p        0.137276  1.000000  0.079317  0.122648  0.224470
degree_t        0.596300  0.079317  1.000000  0.105816  0.011509
workex          0.007856  0.122648  0.105816  1.000000  0.056735
etest_p         0.075643  0.224470  0.011509  0.056735  1.000000
specialisation  0.172107 -0.218286  0.084361 -0.191174 -0.236315
mba_p           0.039345  0.402364  0.116666  0.168811  0.218055
status          0.033442  0.479861 -0.020352  0.276060  0.127639
salary          0.074322 -0.019272  0.169655  0.136920  0.178307

                specialisation     mba_p    status    salary
sl_no                 0.046630  0.022327 -0.026859  0.063764
gender               -0.106160 -0.300531  0.090670  0.158912
ssc_p                -0.172536  0.388478  0.607889  0.035330
ssc_b                -0.051565  0.083120  0.037297  0.005539
hsc_p                -0.241630  0.354823  0.491228  0.076819
hsc_b                 0.002232  0.090201  0.016945 -0.007549
hsc_s                 0.172107  0.039345  0.033442  0.074322
degree_p             -0.218286  0.402364  0.479861 -0.019272
degree_t              0.084361  0.116666 -0.020352  0.169655
workex               -0.191174  0.168811  0.276060  0.136920
etest_p              -0.236315  0.218055  0.127639  0.178307
specialisation        1.000000 -0.105728 -0.250655 -0.146576
mba_p                -0.105728  1.000000  0.076922  0.175013
status               -0.250655  0.076922  1.000000       NaN
salary               -0.146576  0.175013       NaN  1.000000
```

[47]:

```
import seaborn as asns
sns.heatmap(df.corr(),annot=False)
```

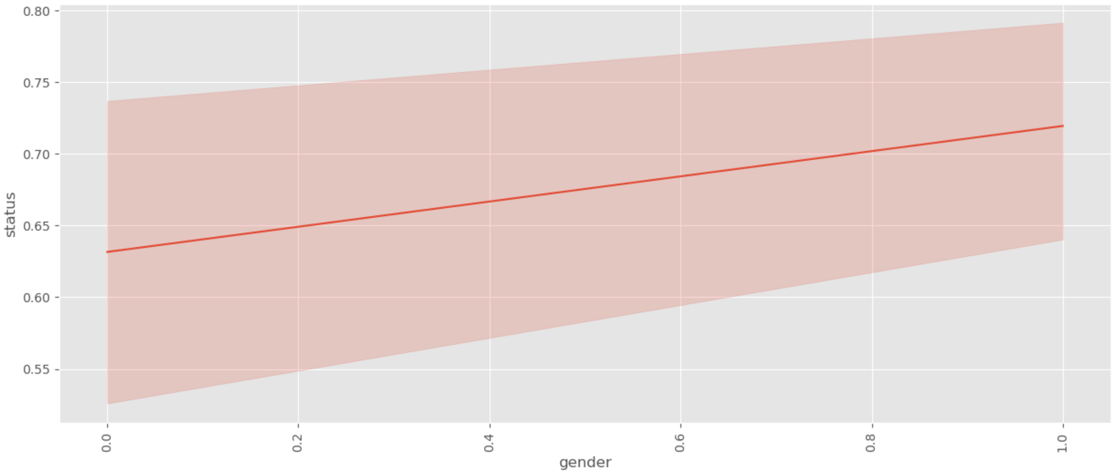
[47]:

```
<AxesSubplot:>
```



[49]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(15,6))
sns.lineplot(y = df['status'], x = df['gender'])
plt.xticks(rotation = 90)

plt.show()
```



[50]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sl_no           215 non-null    int64
 1   gender          215 non-null    int32
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    int32
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    int32
 6   hsc_s           215 non-null    int32
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    int32
 9   workex          215 non-null    int32
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    int32
 12  mba_p           215 non-null    float64
 13  status          215 non-null    int32
 14  salary          148 non-null    float64
dtypes: float64(6), int32(8), int64(1)
memory usage: 18.6 KB
```

[55]:
```
df['salary'].dtype
```

[55]:
```
dtype('float64')
```

[51]:
```
import pandas as pd
import numpy as np
```

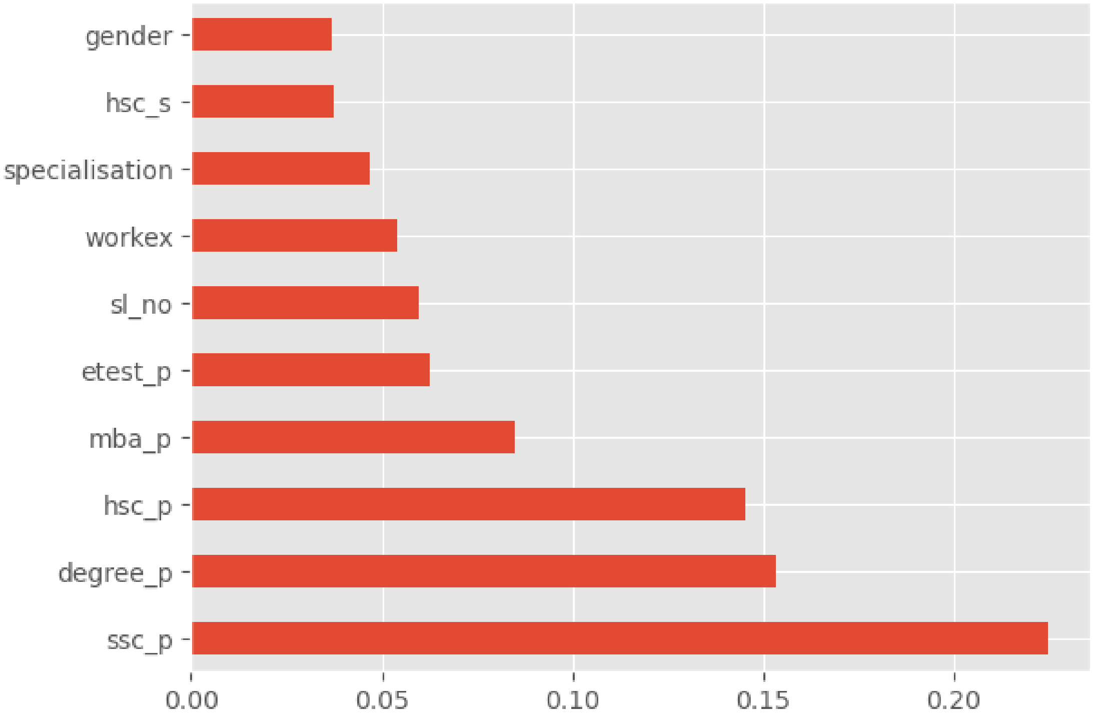
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
X = df.iloc[:,0:13]  #independent columns
y = df.iloc[:,-2]    #target column i.e price range#apply SelectKBest class to␣
 ↪extract top 10 best features



bestfeatures = SelectKBest(score_func=chi2, k=10)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score']  #naming the dataframe columns
print(featureScores.nlargest(10,'Score'))
```

```
             Specs       Score
2            ssc_p  137.739258
4            hsc_p   92.449312
7         degree_p   40.204896
9           workex   10.745484
10         etest_p    8.522679
11  specialisation    7.539357
0            sl_no    5.532155
12           mba_p    0.691857
1           gender    0.624805
3            ssc_b    0.161359
```

[52]:

```
import pandas as pd
import numpy as np

X = df.iloc[:,0:13]  #independent columns
y = df.iloc[:,-2]    #target column i.e price range
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X,y)
print(model.feature_importances_) #use inbuilt class feature_importances of␣
 ↪tree based classifiers
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```

```
[0.05971068 0.03710311 0.22471598 0.03283039 0.14518795 0.03152152
 0.03748728 0.15332656 0.03017956 0.05386365 0.06258802 0.04664588
 0.08483942]
```



[68]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sl_no           215 non-null    int64
 1   gender          215 non-null    object
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object
 6   hsc_s           215 non-null    object
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object
 9   workex          215 non-null    object
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object
 14  salary          215 non-null    float64
```

```
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

[6]:
```
df=pd.read_csv(r"C:\Users\Ranveer\Downloads\Placement_Data_Full_Class.csv")
```

[7]:
```
df['salary'].fillna(value = 0.0, inplace = True)
```

[71]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sl_no           215 non-null    int64
 1   gender          215 non-null    object
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object
 6   hsc_s           215 non-null    object
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object
 9   workex          215 non-null    object
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object
 14  salary          215 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

[72]:
```
df['salary']
```

[72]:
```
0      270000.0
1      200000.0
2      250000.0
3           0.0
4      425000.0
         …
210    400000.0
211    275000.0
212    295000.0
213    204000.0
214         0.0
Name: salary, Length: 215, dtype: float64
```

[ ]: