ABSTRACT

The project aims to find that, based on the placement history of the students and their academic results, the propensity of the students getting placed for a given academic year and which variables are mainly influencing in student placement. In this project, the exploratory data analysis and further analysis are done using the **Jupyter notebook**. Here, logistic regression is used as a predictive model. The accuracy rate for the predictive model is 0.83 for training data and 0.88 for test data. The confusion metrics is incorporated here. To improve the accuracy, use Random Forest Ensemble learning technique and the accuracy score increased by 0.90. Factors such as Secondary education percentage(ssc_p), Higher secondary percentage(hsc_p), Degree course percentage(degree_p), Specialization, and Work experience are the main factors for a student's placement.

DATA SOURCE, METHODOLOGY, AND VARIABLES

This campus placement data is collected from Kaggle.

sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisat	mba_p	status	salary
1	M	67	Others	91	Others	Commerce	58	Sci&Tech	No	55	Mkt&HR	58.8	Placed	270000
2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000
3	M	65	Central	68	Central	Arts	64	Comm&M	No	75	Mkt&Fin	57.8	Placed	250000
4	M	56	Central	52	Central	Science	52	Sci&Tech	No	66	Mkt&HR	59.43	Not Placed	
5	M	85.8	Central	73.6	Central	Commerce	73.3	Comm&M	No	96.8	Mkt&Fin	55.5	Placed	425000
6	M	55	Others	49.8	Others	Science	67.25	Sci&Tech	Yes	55	Mkt&Fin	51.58	Not Placed	
7	F	46	Others	49.2	Others	Commerce	79	Comm&M	No	74.28	Mkt&Fin	53.29	Not Placed	
8	M	82	Central	64	Central	Science	66	Sci&Tech	Yes	67	Mkt&Fin	62.14	Placed	252000
9	M	73	Central	79	Central	Commerce	72	Comm&M	No	91.34	Mkt&Fin	61.29	Placed	231000
10	M	58	Central	70	Central	Commerce	61	Comm&M	No	54	Mkt&Fin	52.21	Not Placed	
11	M	58	Central	61	Central	Commerce	60	Comm&M	Yes	62	Mkt&HR	60.85	Placed	260000
12	M	69.6	Central	68.4	Central	Commerce	78.3	Comm&M	Yes	60	Mkt&Fin	63.7	Placed	250000
13	F	47	Central	55	Others	Science	65	Comm&M	No	62	Mkt&HR	65.04	Not Placed	
14	F	77	Central	87	Central	Commerce	59	Comm&M	No	68	Mkt&Fin	68.63	Placed	218000
15	M	62	Central	47	Central	Commerce	50	Comm&M	No	76	Mkt&HR	54.96	Not Placed	
16	F	65	Central	75	Central	Commerce	69	Comm&M	Yes	72	Mkt&Fin	64.66	Placed	200000
17	M	63	Central	66.2	Central	Commerce	65.6	Comm&M	Yes	60	Mkt&Fin	62.54	Placed	300000
18	F	55	Central	67	Central	Commerce	64	Comm&M	No	60	Mkt&Fin	67.28	Not Placed	
19	F	63	Central	66	Central	Commerce	64	Comm&M	No	68	Mkt&HR	64.08	Not Placed	
20	M	60	Others	67	Others	Arts	70	Comm&M	Yes	50.48	Mkt&Fin	77.89	Placed	236000
21	M	62	Others	65	Others	Commerce	66	Comm&M	No	50	Mkt&HR	56.7	Placed	265000
22	F	79	Others	76	Others	Commerce	85	Comm&M	No	95	Mkt&Fin	69.06	Placed	393000
23	F	69.8	Others	60.8	Others	Science	72.23	Sci&Tech	No	55.53	Mkt&HR	68.81	Placed	360000
24	F	77.4	Others	60	Others	Science	64.74	Sci&Tech	Yes	92	Mkt&Fin	63.62	Placed	300000
25	M	76.5	Others	97.7	Others	Science	78.86	Sci&Tech	No	97.4	Mkt&Fin	74.01	Placed	360000
26	F	52.58	Others	54.6	Central	Commerce	50.2	Comm&M	Yes	76	Mkt&Fin	65.33	Not Placed	
27	M	71	Others	79	Others	Commerce	66	Comm&M	Yes	94	Mkt&Fin	57.55	Placed	240000
28	M	63	Others	67	Others	Commerce	66	Comm&M	No	68	Mkt&HR	57.69	Placed	265000

The variables are explained below:

- ssc_p = Secondary education percentage
- ssc_b = Board of education
- hsc_p = Higher Secondary education percentage
- hsc_b = Higher Secondary Board of education
- degree_p = Degree percentage
- degree_t = Feild of degree education
- workex = Work experience
- etest_p = Employment test percentage
- specialization = Post grad specialization
- mba_p = MBA Percentage.

Here we use the Jupyter notebook for analysis purposes.

Data analysis

- Data pre-processing and data description:

At the very beginning of the project, we checked the dimension of the dataset.

```
In [4]:  1  placement.shape
Out[4]: (215, 15)
```

After that, we checked the variable types.

```
In [7]:  1  placement.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
```

Here, the continuous variables are ssc_p, hsc_p, degree_p, etest_p, mba_p, salary, and the categorical Variables are gender, ssc_b, hsc_b, hsc_s, degree_t, specialisation, workex, status.

In the next step, we will check the missing and duplicate values.

```
In [91]: 1 placement.isnull().sum()

Out[91]: sl_no             0
         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
         salary           67
         dtype: int64
```

Here we can observe that 67 out of 215 rows in the salary column are missing which is about 31% of the data. So, we can drop this column.

```
In [95]: 1 placement.duplicated().sum()

Out[95]: 0
```

There is no duplicate value present in the data.

Now, we will also check the description as well as the correlation between the variables.

```
In [177]: 1 placement.describe()
```

Out[177]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p
count	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000
mean	108.000000	67.303395	66.333163	66.370186	72.100558	62.278186
std	62.209324	10.827205	10.897509	7.358743	13.275956	5.833385
min	1.000000	40.890000	37.000000	50.000000	50.000000	51.210000
25%	54.500000	60.600000	60.900000	61.000000	60.000000	57.945000
50%	108.000000	67.000000	65.000000	66.000000	71.000000	62.000000
75%	161.500000	75.700000	73.000000	72.000000	83.500000	66.255000
max	215.000000	89.400000	97.700000	91.000000	98.000000	77.890000

In [180]:
```
1 placement.corr()
```

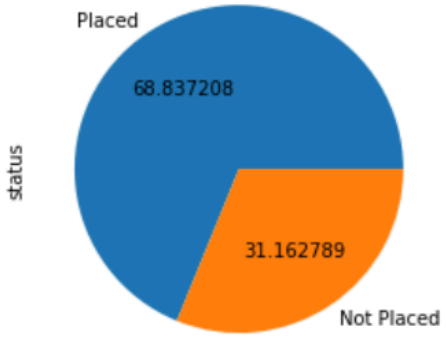
Out[180]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p
sl_no	1.000000	-0.078155	-0.085711	-0.088281	0.063636	0.022327
ssc_p	-0.078155	1.000000	0.511472	0.538404	0.261993	0.388478
hsc_p	-0.085711	0.511472	1.000000	0.434206	0.245113	0.354823
degree_p	-0.088281	0.538404	0.434206	1.000000	0.224470	0.402364
etest_p	0.063636	0.261993	0.245113	0.224470	1.000000	0.218055
mba_p	0.022327	0.388478	0.354823	0.402364	0.218055	1.000000

After completing the pre-processing part, we moved to the univariate analysis.

- Univariate analysis:

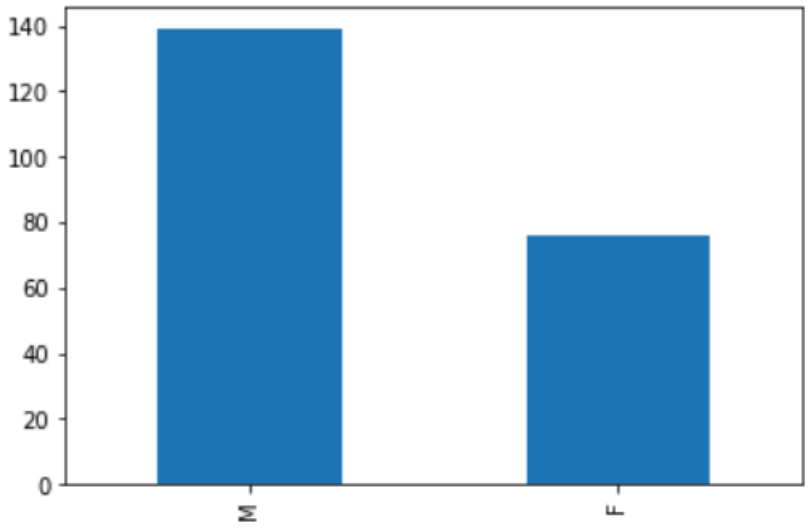
```
In [184]: 1 placement["status"].value_counts().plot(kind="pie",autopct='%2f')
Out[184]: <AxesSubplot:ylabel='status'>
```



Out of 215 students, 68% of students are placed and 31% of the students are not placed.

Now, we will check the male-female ratio. There are 64% of male and 35% are female.

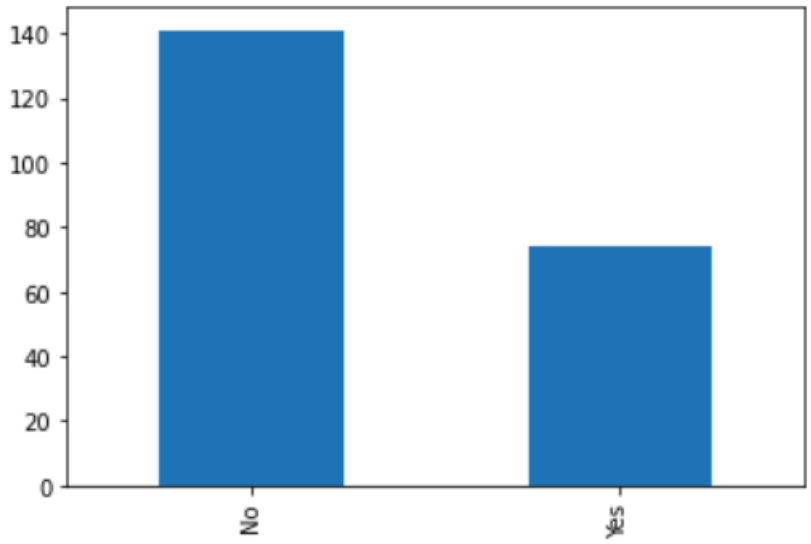
```
In [185]: 1 placement['gender'].value_counts().plot(kind="bar")
          2 placement['gender'].value_counts()
Out[185]: M    139
          F     76
          Name: gender, dtype: int64
```



Next, we checked the work experience amount among the students. 65% of students do not have any work experience and 34% of students have work experience.

In [188]:
```
placement['workex'].value_counts().plot(kind="bar")
placement['workex'].value_counts()
```

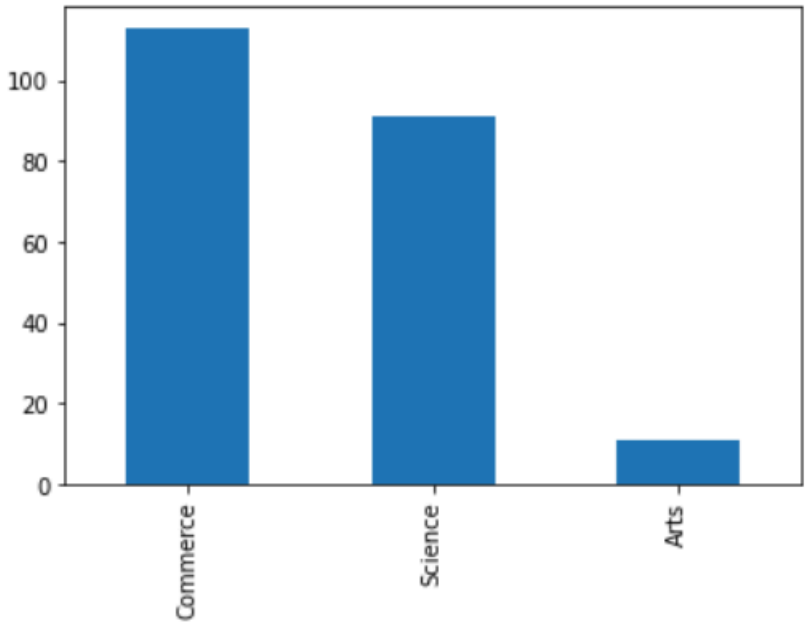
Out[188]:
```
No     141
Yes     74
Name: workex, dtype: int64
```



Next, we check the higher Secondary subject(hsc_s) column, and from the results, we observe that out of 215, 52% of the students belong to a commerce background, 42% belong to a science background, and the rest 5% belong to an arts background.

```
In [189]:
1 placement['hsc_s'].value_counts().plot(kind="bar")
2 placement['hsc_s'].value_counts()
```

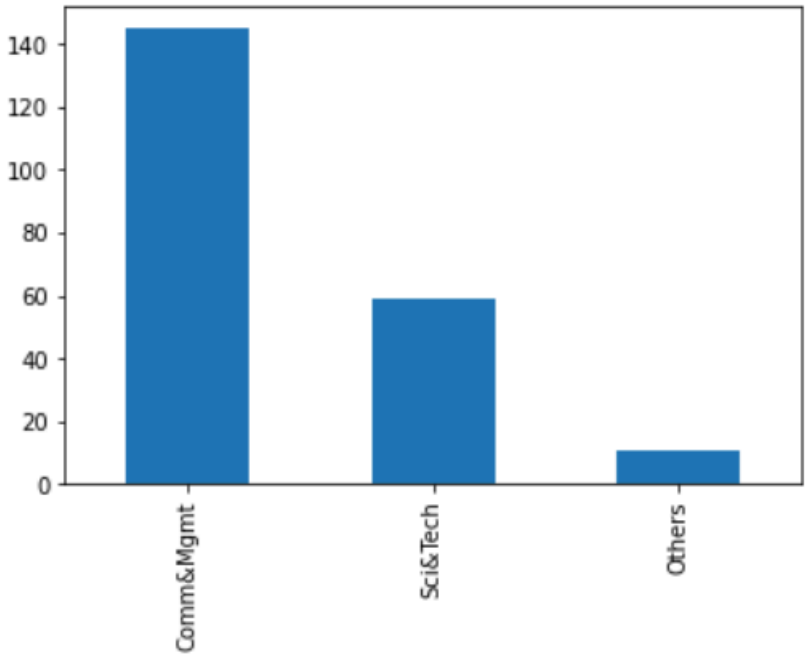
```
Out[189]: Commerce    113
          Science      91
          Arts         11
          Name: hsc_s, dtype: int64
```



Next, we will analyze the degree course type(degree_t) column. Out of 215 students, 67% of students have a degree in commerce and management. Then 27% of students have a degree in science and technology and the rest 5% belong to other degree courses. The noticeable point here, the number of students from the commerce stream increased from a higher secondary level which was 52%, and the decrease in the science stream from the higher secondary level which was 42% to 27% in degree courses.

In [190]:
```
1 placement['degree_t'].value_counts().plot(kind="bar")
2 placement['degree_t'].value_counts()
```

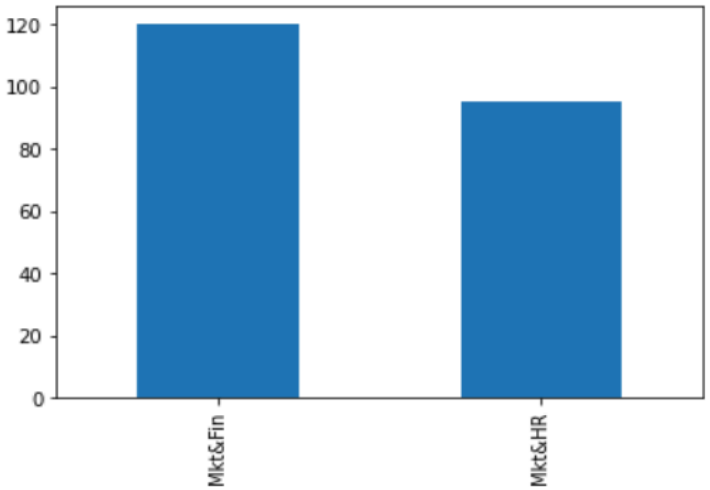
Out[190]:
```
Comm&Mgmt    145
Sci&Tech      59
Others        11
Name: degree_t, dtype: int64
```



We will now check the specializations of the students. Here, 55% of the students have marketing and finance, and the rest 44% of the students have marketing and human resource as specialization in MBA.

```
In [191]:
1 placement['specialisation'].value_counts().plot(kind="bar")
2 placement['specialisation'].value_counts()
```

```
Out[191]: Mkt&Fin    120
          Mkt&HR      95
          Name: specialisation, dtype: int64
```

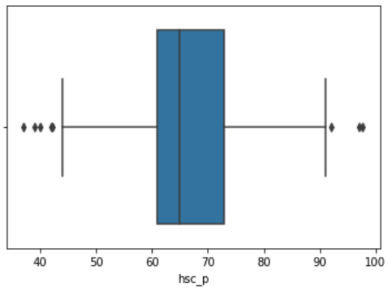


Now, we will check for outliers. Our approach will be the Boxplot. In our dataset, we have 5 continuous variables, which are ssc_p, hsc_p, degree_p, etest_p, and mba_p.

```
In [204]:
1 sns.boxplot(placement['hsc_p'])
```

```
D:\Python\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ver
sion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  warnings.warn(
```

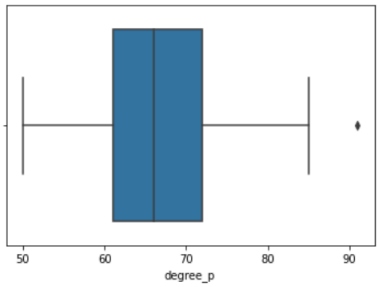
```
Out[204]: <AxesSubplot:xlabel='hsc_p'>
```



In [210]:
```
1 sns.boxplot(placement['degree_p'])
```

```
D:\Python\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ver
sion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  warnings.warn(
```

Out[210]: <AxesSubplot:xlabel='degree_p'>



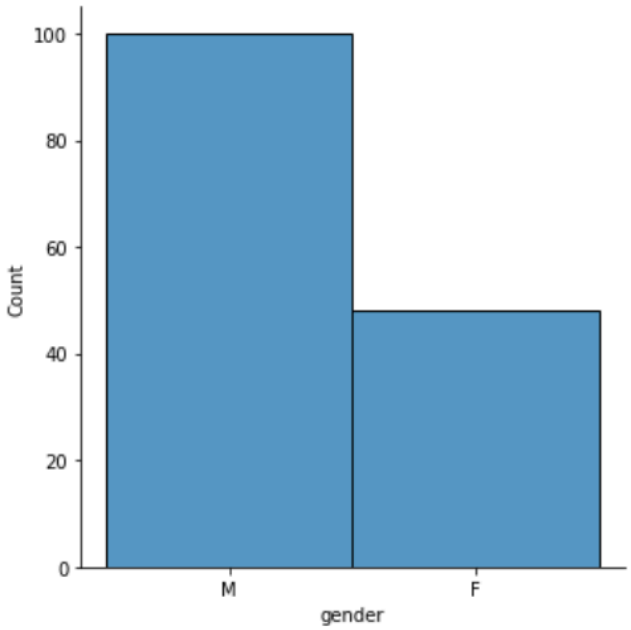
From our boxplot analysis, we found that hsc_p and degree_p are those variables that have outliers. The boxplot gives us a graphical representation of the outliers present in our data. But if we observe closely then, it can be observed that we are analyzing the higher secondary percentage and degree percentage whose value lies between 0 to 100, and here the outliers are lies between this range which means a student can get 99 percentile, so technically these values will not be an outlier. We will consider these values as outliers if these values are greater than 100 then we will consider these as outliers. So, we will not treat them as an outlier.

- Bivariate analysis:

After the univariate analysis, now we move to the bivariate analysis. First, we will make a gender-wise comparison of placement.

```
In [218]:
1  sns.displot(placement[placement["status"]=="Placed"]['gender'])
2  placement[placement["status"]=="Placed"]['gender'].value_counts()
```

```
Out[218]: M    100
          F     48
          Name: gender, dtype: int64
```

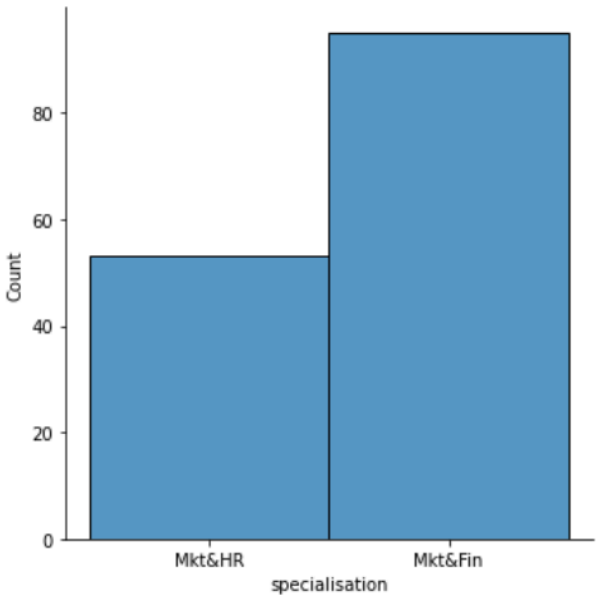


The results show that out of 139 males 100 males are placed, so mele's placement rate is about 71% and out of 76 females 48 females are placed which means the female placement rate is 63%.

Now we will check the placement from the MBA specialization point of view.

```
In [219]:
1 sns.displot(placement[placement["status"]=="Placed"]['specialisation'])
2 placement[placement["status"]=="Placed"]['specialisation'].value_counts()
```

```
Out[219]: Mkt&Fin    95
          Mkt&HR     53
          Name: specialisation, dtype: int64
```

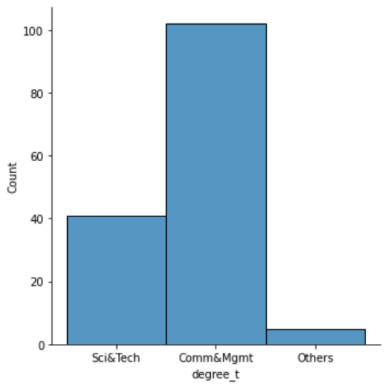


Here out of 120 students from the Marketing and finance specialization, 95 students are placed so the rate is 79%. On the other hand, out of 95 students from the Marketing and human resource specialization 53 students are placed, which means the placement rate is 55%.

Checking placements based on graduation degree.

```
In [220]:
1 sns.displot(placement[placement["status"]=="Placed"]['degree_t'])
2 placement[placement["status"]=="Placed"]['degree_t'].value_counts()
3 #based on the cdistribution of the class in comparison to the response variables we must be able to translate the patterns o
```

```
Out[220]: Comm&Mgmt    102
          Sci&Tech      41
          Others         5
          Name: degree_t, dtype: int64
```

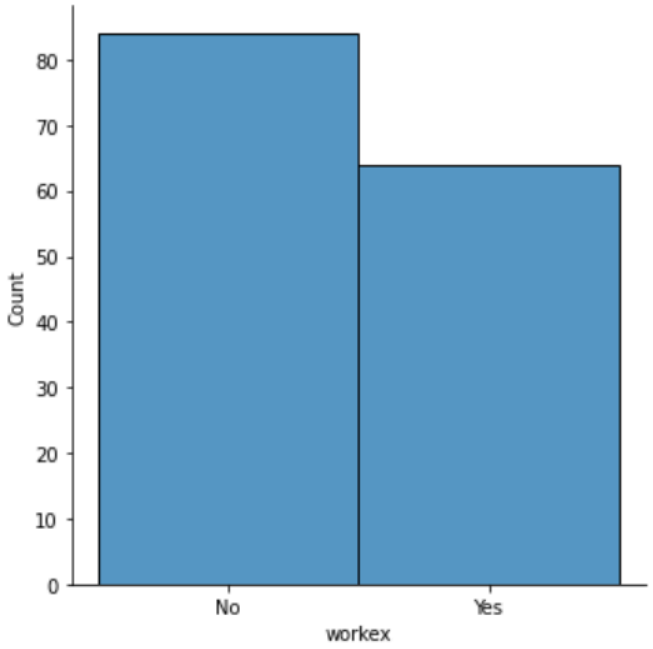


Here we can observe that majority of the placement happens in Commerce and management, the rate is about 70%, and in Science and technology, the placement rate is 69%. So, there is a high possibility for students from commerce and management backgrounds to get placed.

Now, finally, we will check the placements based on work experience.

```
In [222]:
1 sns.displot(placement[placement["status"]=="Placed"]['workex'])
2 placement[placement["status"]=="Placed"]['workex'].value_counts()
```

```
Out[222]: No     84
          Yes    64
          Name: workex, dtype: int64
```



Here, 64 out of 74 students with work experience have been placed, meaning the placement rate with work experience is 86%. On the other hand, out of 141 students who do not have work experience, just 84 students are placed, so the rate is 59%. So, work experience is a major factor in a student's placement.

- Exploratory data analysis in short:

1) Total of 68% of Students are placed while 31% are not placed.
2) Average salary a student gets = 288655 LPA. Minimum and maximum salary = 200000 and 940000 LPA.
3) No. of male students = 139 and No. of female students = 76.
4) 100 boys are placed remaining 39 are not placed --- boys placement rate 71.94%.
5) 48 girls are placed remaining 28 are not placed --- the girl's placement rate is 63.15%. It seems like there is a difference between boys and girls placements but the difference is too little.
6) 74 students have work experience, and out of them 64 are placed, so the workex placement rate is 86.48%, which means work experience important factor in students' placement.

7) 141 students do not have work experience, out of them 84 are placed, so without workex placement rate is 59.57%.
8) Students from commerce and management = 145, placed = 102, rate = 70.34%.
9) Students from Sci&tech = 59, placed = 41, rate = 69.49%.
10) Students from other = 11, placed = 5, rate = 45.45%. Students from a commerce background have a higher possibility to get placed.
11) From the data we can also infer that majority of the recruiters are from the banking and commerce industries.
12) Students from Mkt&Fin = 120, placed = 95, rate = 79.16%. Students from Mkt&HR = 95, placed = 53, rate = 55.78%.

- Feature selection:

Our dependent variable is Status. Now we look at the variables which are mainly influencing the student's placement. In the dataset the Continuous Variables are -- sl_no, ssc_p, hsc_p, degree_p, etest_p, mba_p, salary, and the categorical variables are -- gender, ssc_b, hsc_b, hsc_s, degree_t, workex, specialisation, status (Target). To find the relationship between the Categorical and Continuous variables our approach will be the ANOVA test and for Categorical and Categorical variables the approach will be the Chi-Square test.

From the ANOVA Test, ssc_p, hsc_p, and degree_p are correlated with the dependent variable.

From the Chi-Square Test, work_ex and specialization are the factors that are correlated with each other.

Now we got the factors that are mainly influencing the dependent variable status.

- Machine learning: Training and testing

In this section, we split our data for training and testing purposes.

In [236]:
```
x=placement_data[predictvar].values
y=placement_data[targetvar].values
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, random_state=555)
```

In [237]:
```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(172, 7)
(172,)
(43, 7)
(43,)
```

After completing the steps of splitting the data set, now we applied the logistic regression algorithm to predict from the test data set based on the training data set. The reasons for choosing a logistic regression are ---

1) The data is linearly separable.
2) Logistic regression is less prone to overfitting.
3) The data set is small.

In [239]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
y_pred= model.predict(x_test)
```

```
D:\Python\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

- Logistic regression training model accuracy

In [240]:
```
print(model.score(x_train, y_train))
```

```
0.8372093023255814
```

- Logistic regression test model accuracy

In [241]:
```
print(model.score(x_test,y_test))
```

```
0.8837209302325582
```

We have also implemented the decision tree algorithm.

In [242]:
```
# decision tree
from sklearn.tree import DecisionTreeClassifier
dt_model= DecisionTreeClassifier()
dt_model.fit(x_train, y_train)
```

Out[242]:
```
DecisionTreeClassifier()
```

- Decision tree training model accuracy

In [243]:
```
print(model.score(x_train, y_train))
```

```
0.8372093023255814
```

- Decsion tree test model accuracy

In [244]:
```
y_predict= dt_model.predict(x_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Out[244]:
```
0.8837209302325582
```

Now we check for the confusion matrix,

In [245]:
```
# confusion metrix

from sklearn.metrics import confusion_matrix
cnf_metrix=confusion_matrix(y_test, y_pred)
np.set_printoptions(precision=2)
```

In [246]:
```
cnf_metrix
```

Out[246]:
```
array([[ 8,  4],
       [ 1, 30]], dtype=int64)
```

In [247]:
```
from sklearn import metrics
print('Accuracy Score:',accuracy_score(y_test, y_pred))
print('Error:', 5/43)
print('Precision/Predicted true positive rate:', metrics.precision_score(y_test, y_pred))
print('Recall/True positive rate:', metrics.recall_score(y_test, y_pred))
print('Specifisity:', 8/9)
print('F1 Score:', metrics.f1_score(y_test, y_pred))
print('ROC AUC Score:', metrics.roc_auc_score(y_test, y_pred))
```

```
Accuracy Score: 0.8837209302325582
Error: 0.11627906976744186
Precision/Predicted true positive rate: 0.8823529411764706
Recall/True positive rate: 0.967741935483871
Specifisity: 0.8888888888888888
F1 Score: 0.923076923076923
ROC AUC Score: 0.8172043010752689
```

To increase the accuracy, we have implemented the random forest algorithm.

In [248]:
```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(x_train, y_train)
```

Out[248]:
```
RandomForestClassifier()
```

In [249]:
```
y_pred=rf.predict(x_test)
```

In [250]:
```
accuracy_score(y_test, y_pred)
```

Out[250]:
```
0.9069767441860465
```

Conclusion

The results from this analysis are,
A total of 68% of Students are placed while 31% are not placed. The average salary a student gets is 288655 LPA. Minimum and maximumsalary 200000 and 940000 LPA. The No. of male students 139 and No. of female students 76. 100 boys are placed remaining 39 are not placed --- boys placement rate 71.94%. 48 girls are placed remaining 28 are not placed --- girls' placement rate is 63.15%. It seems like there is a difference between boys' and girls' placement but the difference is too little. 74 students have work experience, and out of them 64 are placed, so the workex placement rate is 86.48%, which means work experience important factor in students' placement. 141 students do not have work experience, and out of them 84 are placed, so without workex placement rate is 59.57%. Students from commerce and management = 145, placed = 102, rate = 70.34%. Students from Sci&tech = 59, placed = 41, rate = 69.49%. Students from other = 11, placed = 5, rate = 45.45%. Students from a commerce background have a higher possibility to get placed. From the data, we can also infer that majority of the recruiters are from the banking and commerce industries. Students from Mkt&Fin = 120, placed = 95, rate = 79.16%. Students from Mkt&HR = 95, placed = 53, rate = 55.78%. From this analysis of the placement data set, the responsible variables for a student's placement are "ssc_p", "hsc_p", "degree_p", "specialisation", and "workex". Based on the training data set our machine can predict the test data set with 0.88% accuracy and using the Random Forest bagging technique our prediction accuracy increased by 0.90%.