Introduction

In the modern educational landscape, the analysis of placement data holds paramount importance for academic institutions and students alike. The Placement Data Analysis project in Python represents a pivotal endeavor aimed at harnessing the power of programming and data analysis techniques to extract meaningful insights from placement records.

This project delves into the realms of data science and analytics, employing Python as the primary tool for data manipulation, visualization, and interpretation. By leveraging Python's rich ecosystem of libraries such as Pandas, NumPy, Matplotlib, and Seaborn, the project offers a comprehensive framework for processing, analyzing, and visualizing placement data.

At its core, the Placement Data Analysis project serves multiple objectives. Firstly, it facilitates the exploration of trends and patterns within placement records, enabling stakeholders to identify key factors influencing students' employability and career trajectories. Through descriptive statistics, trend analysis, and visualization techniques, the project elucidates insights regarding placement rates, salary distributions, industry preferences, geographic trends, and more.

Moreover, the project fosters predictive modeling endeavors, wherein machine learning algorithms are employed to forecast future placement outcomes based on historical data. By implementing regression, classification, or clustering algorithms, the project empowers institutions to make informed decisions regarding curriculum development, career counseling, and campus recruitment strategies.

Furthermore, the project promotes transparency and accountability by offering stakeholders access to interactive dashboards and reports. Through user-friendly interfaces developed using libraries like Plotly and Dash, the project enables stakeholders to interactively explore placement data, generate custom reports, and derive actionable insights in real-time.

In essence, the Placement Data Analysis project in Python epitomizes the convergence of technology, education, and data-driven decision-making. By harnessing the power of Python programming and data analysis techniques, the project catalyzes a paradigm shift in how educational institutions conceptualize, analyze, and optimize placement processes, ultimately empowering students to embark on fulfilling career journeys.

```
import numpy as np
import pandas as pd
```

```
Placement_analysis= pd.read_csv("/content/Placement_Data .zip")
```

```
Placement_analysis.head(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

```
for i in Placement_analysis.columns:
  print(i)
```

```
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
```

```
print("checking rows & columns, Rows={}, columns={}".format(Placement_analysis.shape[0], Placement_analysis.shape[1]))
```

```
checking rows & columns, Rows=215, columns=15
```

```
Placement_analysis.info()
Placement_analysis.dtypes
```

```
<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
```

```
[12] Mean=np.mean(Placement_analysis["ssc_p"])
     Mean
```

```
67.30339534883721
```

```
[18] print(Placement_analysis["ssc_p"].max())
```

```
89.4
```

```
[24] Placement_analysis[Placement_analysis["ssc_p"]>85]
```

	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
4	5	M	85.8	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0
33	34	F	87.0	Others	65.00	Others	Science	81.00	Comm&Mgmt	Yes	88.0	Mkt&Fin	72.78	Placed	260000.0
43	44	M	87.0	Others	87.00	Others	Commerce	68.00	Comm&Mgmt	No	95.0	Mkt&HR	62.90	Placed	300000.0
62	63	F	86.5	Others	64.20	Others	Science	67.40	Sci&Tech	No	59.0	Mkt&Fin	59.69	Placed	240000.0
145	146	M	89.4	Others	65.66	Others	Science	71.25	Sci&Tech	No	72.0	Mkt&HR	63.23	Placed	400000.0
160	161	M	87.0	Central	74.00	Central	Science	65.00	Sci&Tech	Yes	75.0	Mkt&HR	72.00	Placed	300000.0

```
[26] print(Placement_analysis["degree_t"].unique())
```

```
['Sci&Tech' 'Comm&Mgmt' 'Others']
```

```
Placement_analysis.dropna(how="all")
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN

```
missing_values = Placement_analysis.isnull().sum()
print(missing_values)
```

```
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
dtype: int64
```

```
missing_values = Placement_analysis.isnull().sum()
print(missing_values)
```

```
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
```

```
missing_percentage = (missing_values / len(Placement_analysis)) * 100
print(missing_percentage)
```

```
sl_no              0.000000
gender             0.000000
ssc_p              0.000000
ssc_b              0.000000
hsc_p              0.000000
hsc_b              0.000000
hsc_s              0.000000
degree_p           0.000000
degree_t           0.000000
workex             0.000000
etest_p            0.000000
specialisation     0.000000
mba_p              0.000000
status             0.000000
salary            31.162791
dtype: float64
```

```
threshold= 0
columns_to_remove = missing_percentage[missing_percentage > threshold].index
print(columns_to_remove)
```

```
Index(['salary'], dtype='object')
```

```
df = Placement_analysis.drop(columns=columns_to_remove)
print(df)
```

```
     sl_no gender  ssc_p    ssc_b  hsc_p    hsc_b     hsc_s  degree_p  \
0        1      M  67.00   Others  91.00   Others  Commerce     58.00
1        2      M  79.33  Central  78.33   Others   Science     77.48
2        3      M  65.00  Central  68.00  Central      Arts     64.00
3        4      M  56.00  Central  52.00  Central   Science     52.00
4        5      M  85.80  Central  73.60  Central  Commerce     73.30
..     ...    ...    ...      ...    ...      ...       ...       ...
210    211      M  80.60   Others  82.00   Others  Commerce     77.60
211    212      M  58.00   Others  60.00   Others   Science     72.00
212    213      M  67.00   Others  67.00   Others  Commerce     73.00
213    214      F  74.00   Others  66.00   Others  Commerce     58.00
214    215      M  62.00  Central  58.00   Others   Science     53.00

          degree_t workex  etest_p specialisation  mba_p      status
    0     Sci&Tech     No     55.0         Mkt&HR  58.80      Placed
    1     Sci&Tech    Yes     86.5        Mkt&Fin  66.28      Placed
    2    Comm&Mgmt     No     75.0        Mkt&Fin  57.80      Placed
    3     Sci&Tech     No     66.0         Mkt&HR  59.43  Not Placed
    4    Comm&Mgmt     No     96.8        Mkt&Fin  55.50      Placed
    ..         ...    ...      ...            ...    ...         ...
    210  Comm&Mgmt     No     91.0        Mkt&Fin  74.49      Placed
    211   Sci&Tech     No     74.0        Mkt&Fin  53.62      Placed
    212  Comm&Mgmt    Yes     59.0        Mkt&Fin  69.72      Placed
    213  Comm&Mgmt     No     70.0         Mkt&HR  60.23      Placed
    214  Comm&Mgmt     No     89.0         Mkt&HR  60.22  Not Placed

    [215 rows x 14 columns]
```

```
Placement_analysis.dropna(how="all")
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN

215 rows × 15 columns


```
Placement_analysis.fillna(0)
```

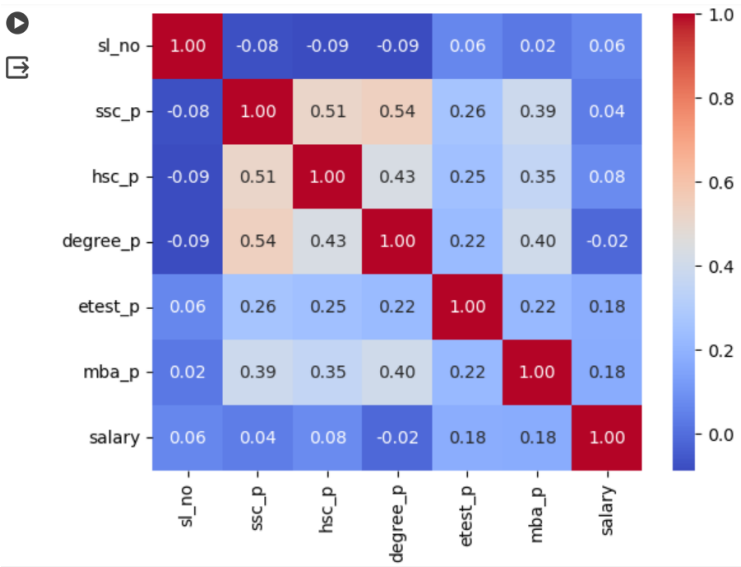
	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	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	0.0
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	0.0

215 rows × 15 columns


```
correlation_matrix = Placement_analysis.corr()
```

```
<ipython-input-13-393a0d1278d8>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will defa
  correlation_matrix = Placement_analysis.corr()
```

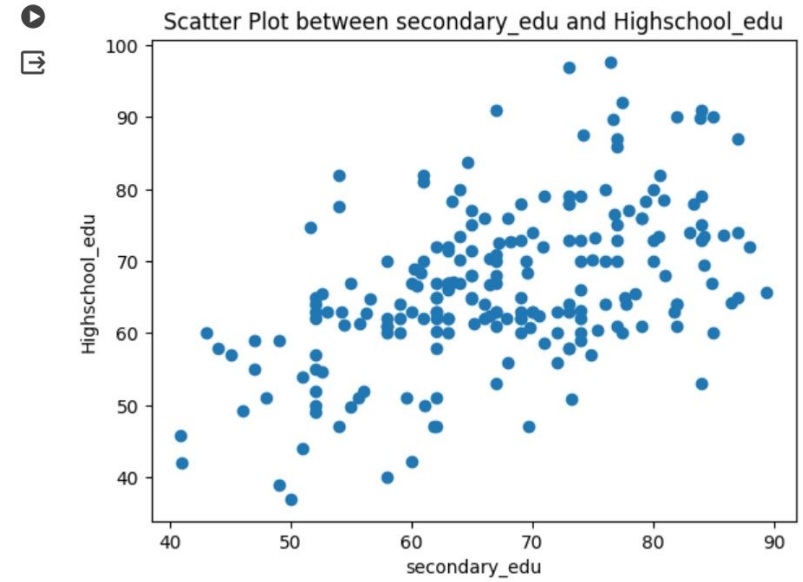
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.show()
```




```
Secondary_edu = 'ssc_p'
Highschool_edu = 'hsc_p'

# Use the corr method
correlation = Placement_analysis[Secondary_edu].corr(Placement_analysis[Highschool_edu])

print(f'Correlation between {Secondary_edu} and {Highschool_edu}: {correlation}')
```

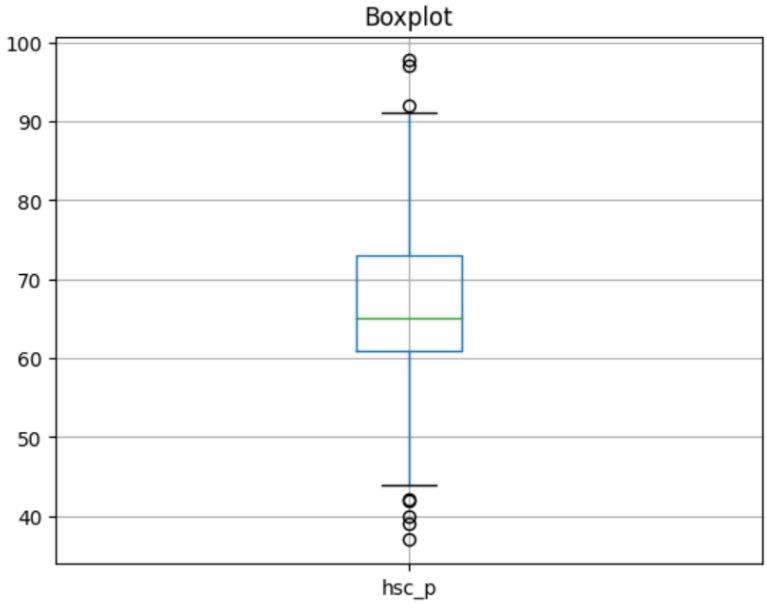
```
Correlation between ssc_p and hsc_p: 0.5114721015997723
```

```
# Create Scatter Plot
plt.scatter(Placement_analysis['ssc_p'], Placement_analysis['hsc_p'])

# Add Labels and Title
plt.xlabel('secondary_edu')
plt.ylabel('Highschool_edu')
plt.title('Scatter Plot between secondary_edu and Highschool_edu')

# Display the Plot
plt.show()
```

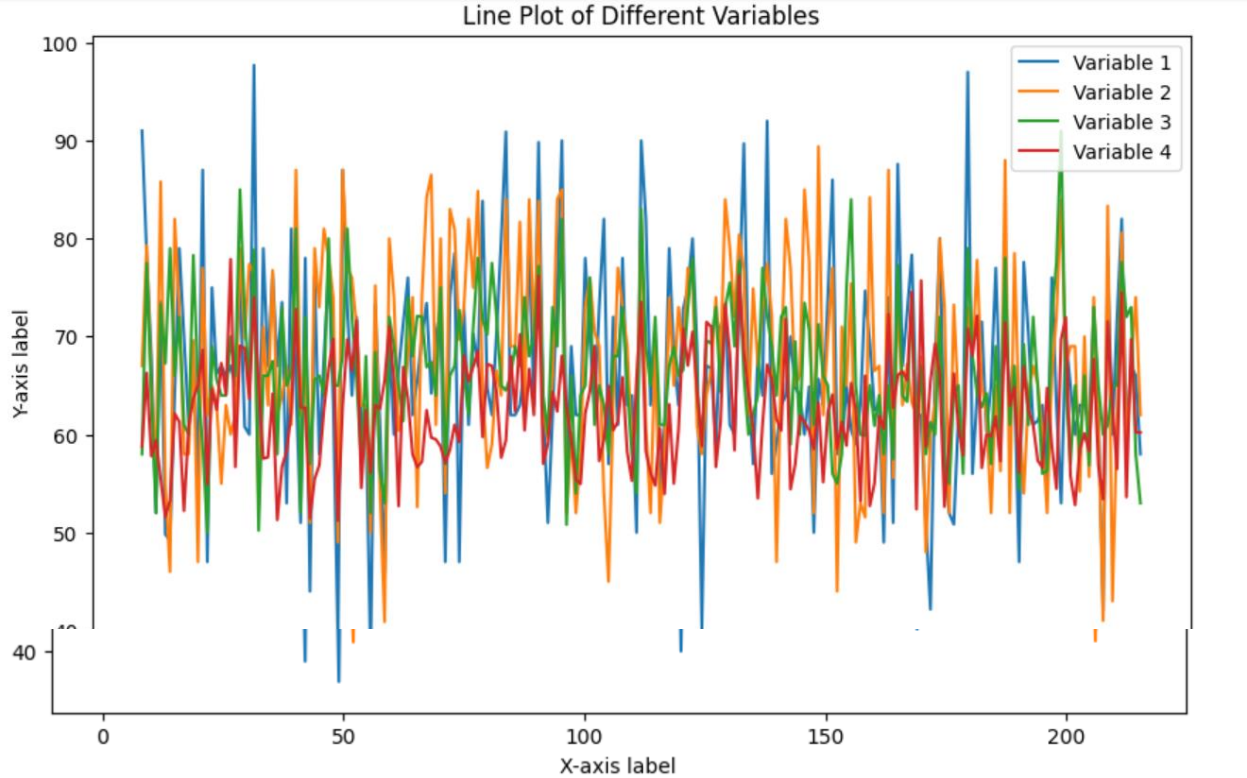


```
[17] import seaborn as sns
```

```
Placement_analysis.boxplot(column='hsc_p')
plt.title('Boxplot')
plt.show()
```




```
x1_values = Placement_analysis['hsc_p']
x2_values = Placement_analysis['ssc_p']
y1_values = Placement_analysis['degree_p']
y2_values = Placement_analysis['mba_p']
# ... add more variables if needed

# Plotting the line plot
plt.figure(figsize=(10, 6))  # Adjust the figure size as needed

# Plot the lines for each variable
plt.plot(x1_values, label='Variable 1')
plt.plot(x2_values, label='Variable 2')
plt.plot(y1_values, label='Variable 3')
plt.plot(y2_values, label='Variable 4')
# ... add more lines for additional variables

# Customize the plot (optional)
plt.title('Line Plot of Different Variables')
plt.xlabel('X-axis label')
plt.ylabel('Y-axis label')
plt.legend()  # Show legend

# Show the plot
plt.show()
```



- ssc_p and hsc_p: There is a very strong positive correlation between ssc_p and hsc_p. This means that students who score high on their SSC exams are also likely to score high on their HSC exams, and vice versa.
- ssc_p and degree_p: There is a very strong positive correlation between ssc_p and degree_p. This means that students who score high on their SSC exams are also likely to score high on their degree exams, and vice versa.
- ssc_p and mba_p: There is a very strong positive correlation between ssc_p and mba_p. This means that students who score high on their SSC exams are also likely to score high on their MBA exams, and vice versa.
- hsc_p and degree_p: There is a very strong positive correlation between hsc_p and degree_p. This means that students who score high on their HSC exams are also likely to score high on their degree exams, and vice versa.
- hsc_p and mba_p: There is a very strong positive correlation between hsc_p and mba_p. This means that students who score high on their HSC exams are also likely to score high on their MBA exams, and vice versa.
- degree_p and mba_p: There is a very strong positive correlation between degree_p and mba_p. This means that students who score high on their degree exams are also likely to score high on their MBA exams, and vice versa.

Overall, these correlations suggest that there is a strong relationship between performance on school exams and performance on higher education entrance exams.

This is likely due to a number of factors, such as the fact that these exams often test similar skills and knowledge. Additionally, students who are successful in school are often more likely to be motivated and disciplined, which can also lead to success in higher education.

Dataset

https://drive.google.com/file/d/1M0SOPww1w2qIebfJbPHwuVuxIra9SIHO/view?usp=drive_link

Conclusion

In conclusion, the Placement Data Analysis project in Python represents a significant milestone in the realm of educational analytics. Through its comprehensive framework for data manipulation, analysis, and visualization, the project has enabled stakeholders to glean valuable insights into placement trends, student employability factors, and future career trajectories.