Important libraries

1. **os**: Provides a way to interact with the operating system for file and directory manipulation.
2. **numpy**: crucial for efficient numerical computing in Python, providing support for large, multi-dimensional arrays and matrices along with a collection of mathematical functions to operate on these arrays.
3. **pandas**: Offers powerful data structures and data analysis tools for handling and analyzing structured data.
4. **matplotlib**: Enables the creation of static, animated, and interactive visualizations in Python.
5. **seaborn**: Simplifies the creation of informative and attractive statistical graphics built on top of matplotlib.
6. **scikit-learn**: Provides a comprehensive suite of tools for machine learning and data mining, including classification, regression, clustering, and dimensionality reduction.

```
# Importing the required libraries and packages
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor,
RandomForestRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import ElasticNet
```

Data Collection

The project will use the **UDISE+ (2021-2022)** data set, developed at the *Department of School Education, Ministry of Education, Government of India*, and maintained by the *National Informatics Centre, Government of India*. This data set includes comprehensive information on school infrastructure, digital facilities, enrollment, and other relevant metrics across India.

The data tables relevant to the project topic are shortlisted and combined into an excel file of several sheets, uploaded on to Kaggle, named **Infrastructure_Technology_Education_India.xlsx** under the dataset **eduinfratechdataindia**.

This collected and shortlisted dataset contains the following data tables:

- school_no_tbl : quantitative data for schools
- enroll_tbl : student enrollment data
- prom_tbl : quantitative data for promotion
- rep_tbl : quantitative data for repetition
- drop_tbl : quantitative data for dropout
- trans_tbl : quantitative data for transition
- projector_tbl : quantitative data for projector availablity
- smart_class_tbl : quantitative data for smart class availablity
- dig_lib_tbl : quantitative data for digital library availablity
- infratech1_tbl : quantitative data for availablity a set of infratech facilities
- infratech2_tbl : quantitative data for availablity a set of infratech facilities

The following code shows how the original dataset is taken as input and prepared for further cleaning and transformation.

```
# Checking for the input dataset in the input directory
print(os.listdir('../input'))
```

```
['Infrastructure_Technology_Education_India.xlsx']
```

```
# Accessing the input dataset and checking for the input data tables
included
raw_set_path='../input/Infrastructure_Technology_Education_India.xlsx'
raw_set=pd.read_excel(raw_set_path,sheet_name=None)
print(raw_set.keys())
```

```
dict_keys(['school_no_tbl', 'enroll_tbl', 'prom_tbl', 'drop_tbl',
'trans_tbl', 'rep_tbl', 'projector_tbl', 'smart_class_tbl',
'dig_lib_tbl', 'infratech1_tbl', 'infratech2_tbl'])
```

```
# Extracting the different sheet (data tables) into different
dataframes for data cleaning and transformation
school_no_tbl=pd.read_excel(raw_set_path,sheet_name='school_no_tbl')
print('school_no_tbl:',school_no_tbl.columns.tolist())
enroll_tbl=pd.read_excel(raw_set_path,sheet_name='enroll_tbl')
print('enroll_tbl:',enroll_tbl.columns.tolist())
prom_tbl=pd.read_excel(raw_set_path,sheet_name='prom_tbl')
print('prom_tbl:',prom_tbl.columns.tolist())
rep_tbl=pd.read_excel(raw_set_path,sheet_name='rep_tbl')
print('rep_tbl:',rep_tbl.columns.tolist())
drop_tbl=pd.read_excel(raw_set_path,sheet_name='drop_tbl')
print('drop_tbl:',drop_tbl.columns.tolist())
trans_tbl=pd.read_excel(raw_set_path,sheet_name='trans_tbl')
print('trans_tbl:',trans_tbl.columns.tolist())
projector_tbl=pd.read_excel(raw_set_path,sheet_name='projector_tbl')
print('projector_tbl:',projector_tbl.columns.tolist())
smart_class_tbl=pd.read_excel(raw_set_path,sheet_name='smart_class_tbl
')
print('smart_class_tbl:',smart_class_tbl.columns.tolist())
```

```
dig_lib_tbl=pd.read_excel(raw_set_path,sheet_name='dig_lib_tbl')
print('dig_lib_tbl:',dig_lib_tbl.columns.tolist())
infratech1_tbl=pd.read_excel(raw_set_path,sheet_name='infratech1_tbl')
print('infratech1_tbl:',infratech1_tbl.columns.tolist())
infratech2_tbl=pd.read_excel(raw_set_path,sheet_name='infratech2_tbl')
print('infratech2_tbl:',infratech2_tbl.columns.tolist())
```

```
school_no_tbl: ['India/ State /UT', 'Total', 'Primary', 'Upper
Primary', 'Secondary', 'Higher Secondary']
enroll_tbl: ['India/ State /UT', 'Total', 'Pre- Primary', 'Primary (1
to 5)', 'Upper Primary\n(6-8)', 'Elementary (1-8)', 'Secondary (9-
10)', 'Higher Secondary\n(11-12)']
prom_tbl: ['India/ State /UT', 'P Boys', 'P Girls', 'P Total', 'UP
Boys', 'UP Girls', 'UP Total', 'S Boys', 'S Girls', 'S Total']
rep_tbl: ['India/ State /UT', 'P Boys', 'P Girls', 'P Total', 'UP
Boys', 'UP Girls', 'UP Total', 'S Boys', 'S Girls', 'S Total']
drop_tbl: ['India/ State /UT', 'P Boys', 'P Girls', 'P Total', 'UP
Boys', 'UP Girls', 'UP Total', 'S Boys', 'S Girls', 'S Total']
trans_tbl: ['India/ State /UT', 'P Boys', 'P Girls', 'P Total', 'UP
Boys', 'UP Girls', 'UP Total', 'S Boys', 'S Girls', 'S Total']
projector_tbl: ['India/ State /UT', 'TS All management', 'TS
Government', 'TS Government Aided', 'TS Pvt. Unaided', 'TS Others',
'All management (N)', 'Government (N)', 'Government Aided (N)', 'Pvt.
Unaided (N)', 'Others (N)', 'All management (%)', 'Government (%)',
'Government Aided (%)', 'Pvt. Unaided (%)', 'Others (%)']
smart_class_tbl: ['India/ State /UT', 'TS All management', 'TS
Government', 'TS Government Aided', 'TS Pvt. Unaided', 'TS Others',
'All management (N)', 'Government (N)', 'Government Aided (N)', 'Pvt.
Unaided (N)', 'Others (N)', 'All management (%)', 'Government (%)',
'Government Aided (%)', 'Pvt. Unaided (%)', 'Others (%)']
dig_lib_tbl: ['India/ State /UT', 'TS All management', 'TS
Government', 'TS Government Aided', 'TS Pvt. Unaided', 'TS Others',
'All management (N)', 'Government (N)', 'Government Aided (N)', 'Pvt.
Unaided (N)', 'Others (N)', 'All management (%)', 'Government (%)',
'Government Aided (%)', 'Pvt. Unaided (%)', 'Others (%)']
infratech1_tbl: ['India/ State /UT', 'Total', 'Library/ Book Bank/
Reading Corner', 'Playground', 'Digital Library', 'Kitchen Garden',
"Girls' Toilet", "Functional Girls' Toilet", "Boys' Toilet",
"Functional Boys' Toilet", 'Electricity', 'Functional Electricity',
'Solar Panel']
infratech2_tbl: ['India/ State /UT', 'Total', 'Computers used for
pedagogical purposes', 'functional Computers used for pedagogical
purposes', 'Internet Facility', 'Drinking Water', 'Functional Drinking
Water', 'Hand wash facility', 'Rainwater Harvesting System',
'Conducting Medical Checkup of Students in Last Academic Year',
'Ramp', 'Ramp and Handrails', 'Schools with CWSN\nToilet facilities']
```

Data Cleaning and Transformation

Since the data is collected from reliable government sources, there is no need to handle missing values, outliers, and inconsistencies in the data. This can be verified at last after obtaining the final dataset.

Apart from that, we do need to eliminate unnecessary data columns (that have no significance in our study), rename the columns as per requirement and convinience, standardize data formats, and integrate the different tables into a final dataset;

Dataframe : **'final_set'** *Output File* : **'modified_file.xlsx'**

This final dataset is then checked for any missing values or data inconsistencies.

Cleaning and Transformation

```
#school_no_tbl
school_no_tbl=school_no_tbl.drop(school_no_tbl.columns[2:],axis=1)
print('school_no_tbl:',school_no_tbl.columns.tolist())
#enroll_tbl
enroll_tbl=enroll_tbl.drop(enroll_tbl.columns[2:],axis=1)
enroll_tbl.rename(columns={'Total':'Enrollment'},inplace=True)
print('enroll_tbl:',enroll_tbl.columns.tolist())
#prom_tbl
prom_tbl['Promotion Rate'] = prom_tbl[['P Total', 'UP Total', 'S
Total']].mean(axis=1)
prom_tbl=prom_tbl[['India/ State /UT','Promotion Rate']]
print('prom_tbl:',prom_tbl.columns.tolist())
#drop_tbl
drop_tbl['Dropout Rate'] = drop_tbl[['P Total','UP Total','S
Total']].mean(axis=1)
drop_tbl=drop_tbl[['India/ State /UT','Dropout Rate']]
print('drop_tbl:',drop_tbl.columns.tolist())
#trans_tbl
trans_tbl['Transition Rate'] = trans_tbl[['P Total','UP Total','S
Total']].mean(axis=1)
trans_tbl=trans_tbl[['India/ State /UT','Transition Rate']]
print('trans_tbl:',trans_tbl.columns.tolist())
#rep_tbl
rep_tbl['Repetition Rate'] = rep_tbl[['P Total','UP Total','S
Total']].mean(axis=1)
rep_tbl=rep_tbl[['India/ State /UT','Repetition Rate']]
print('rep_tbl:',rep_tbl.columns.tolist())
#projector_tbl
projector_tbl=projector_tbl[['India/ State /UT','All management (%)']]
projector_tbl.rename(columns={'All management
(%)':'Projector'},inplace=True)
print('projector_tbl:',projector_tbl.columns.tolist())
#smart_class_tbl
smart_class_tbl=smart_class_tbl[['India/ State /UT','All management
```

```
(%)']]
smart_class_tbl.rename(columns={'All management (%)':'Smart
Class'},inplace=True)
print('smart_class_tbl:',smart_class_tbl.columns.tolist())
#dig_lib_tbl
dig_lib_tbl=dig_lib_tbl[['India/ State /UT','All management (%)']]
dig_lib_tbl.rename(columns={'All management (%)':'Digital
Library'},inplace=True)
print('dig_lib_tbl:',dig_lib_tbl.columns.tolist())
#infratech1_tbl
infratech1_tbl=infratech1_tbl.drop(infratech1_tbl.columns[[4,6,8,10,12
]],axis=1)
numeric_cols =
infratech1_tbl.select_dtypes(include=['number']).columns
infratech1_tbl[numeric_cols] =
infratech1_tbl[numeric_cols].astype(float)
for i in range(2,8):

infratech1_tbl.iloc[:,i]=(infratech1_tbl.iloc[:,i]/infratech1_tbl.iloc
[:,1])*100
infratech1_tbl=infratech1_tbl.drop(infratech1_tbl.columns[[1]],axis=1)
infratech1_tbl.rename(columns={'Library/ Book Bank/ Reading
Corner':'Library',"Functional Girls' Toilet":"Girls'
Toilet","Functional Boys' Toilet":"Boys' Toilet","Functional
Electricity":"Electricity"},inplace=True)
print('infratech1_tbl:',infratech1_tbl.columns.tolist())
#infratech2_tbl
infratech2_tbl=infratech2_tbl.drop(infratech2_tbl.columns[[2,5,8,9]],a
xis=1)
numeric_cols =
infratech2_tbl.select_dtypes(include=['number']).columns
infratech2_tbl[numeric_cols] =
infratech2_tbl[numeric_cols].astype(float)
for i in range(2,9):

infratech2_tbl.iloc[:,i]=(infratech2_tbl.iloc[:,i]/infratech2_tbl.iloc
[:,1])*100
infratech2_tbl=infratech2_tbl.drop(infratech2_tbl.columns[[1]],axis=1)
infratech2_tbl.rename(columns={'functional Computers used for
pedagogical purposes':'Computers',"Hand wash facility":"Hand
wash","Functional Drinking Water":"Drinking Water","Internet
Facility":"Internet","Schools with CWSN\nToilet facilities":"CWSN
Toilet"},inplace=True)
print('infratech2_tbl:',infratech2_tbl.columns.tolist())
```

```
school_no_tbl: ['India/ State /UT', 'Total']
enroll_tbl: ['India/ State /UT', 'Enrollment']
prom_tbl: ['India/ State /UT', 'Promotion Rate']
drop_tbl: ['India/ State /UT', 'Dropout Rate']
trans_tbl: ['India/ State /UT', 'Transition Rate']
```

```
rep_tbl: ['India/ State /UT', 'Repetition Rate']
projector_tbl: ['India/ State /UT', 'Projector']
smart_class_tbl: ['India/ State /UT', 'Smart Class']
dig_lib_tbl: ['India/ State /UT', 'Digital Library']
infratech1_tbl: ['India/ State /UT', 'Library', 'Playground', 'Kitchen
Garden', "Girls' Toilet", "Boys' Toilet", 'Electricity']
infratech2_tbl: ['India/ State /UT', 'Computers', 'Internet',
'Drinking Water', 'Hand wash', 'Ramp', 'Ramp and Handrails', 'CWSN
Toilet']
```

Merging of all Data tables

```
# Merging the transformed data into one final dataframe
final_set=school_no_tbl
tbl_list=[enroll_tbl,prom_tbl,drop_tbl,trans_tbl,rep_tbl,projector_tbl
,smart_class_tbl,dig_lib_tbl,infratech1_tbl,infratech2_tbl]
for tbl in tbl_list:
    final_set=pd.merge(final_set,tbl, on='India/ State /UT')
print(final_set)
```

```
                India/ State /UT    Total  Enrollment  Promotion
Rate  \
0                          India  1489115   265235830       93.566667

1   Andaman and Nicobar Islands      416       73861       97.833333

2                 Andhra Pradesh    61948     8244647       94.000000

3              Arunachal Pradesh     3603      354382       89.100000

4                          Assam    60859     7544960       87.766667

5                          Bihar    93165    27472692       91.600000

6                     Chandigarh      233      268627      100.000000

7                   Chhattisgarh    56512     5992197       94.933333

8                          Delhi     5619     4572107       97.666667

9                            Goa     1510      304982       96.500000

10                       Gujarat    53851    11542276       92.433333

11                       Haryana    23726     6035679       97.633333

12               Himachal Pradesh    18028     1437022       99.300000

13              Jammu and Kashmir    28805     2718644       95.533333
```

14	Jharkhand	44855	7970750	94.766667
15	Karnataka	76450	12092381	93.366667
16	Kerala	16240	6423120	98.166667
17	Ladakh	978	59788	95.600000
18	Lakshadweep	38	13586	99.000000
19	Madhya Pradesh	125582	16169265	91.933333
20	Maharashtra	109605	22586695	95.766667
21	Manipur	4617	693194	93.166667
22	Meghalaya	14600	1169720	83.600000
23	Mizoram	3911	309904	92.333333
24	Nagaland	2718	443796	88.533333
25	Odisha	62291	7576893	88.333333
26	Puducherry	736	255546	95.866667
27	Punjab	27701	6147500	91.033333
28	Rajasthan	106373	17667510	94.366667
29	Sikkim	1259	135963	94.666667
30	Tamil Nadu	58801	12830951	98.433333
31	Telangana	43083	6915241	94.400000
32	Tripura	4929	713862	94.766667
33	Uttar Pradesh	258054	47181438	93.966667
34	Uttarakhand	22815	2449926	96.933333
35	West Bengal	94744	18733367	88.133333

	Dropout Rate	Transition Rate	Repetition Rate	Projector	Smart Class
0	5.700000	86.800000	0.766667	16.7	14.9
1	2.133333	98.333333	0.066667	27.6	37.3

```
2       5.966667        89.000000        0.166667       23.8
18.0
3       9.233333        89.600000        1.700000       22.4
9.3
4      11.700000        82.266667        0.533333        5.9
2.4
5       8.366667        74.066667        0.066667        3.2
4.6
6       0.000000       107.666667        0.366667       85.0
41.2
7       4.866667        90.900000        0.233333        9.3
5.9
8       1.600000        98.500000        0.733333       60.8
43.6
9       3.000000        96.766667        0.533333       41.1
9.3
10      7.633333        85.700000        0.000000       38.5
21.9
11      2.033333        96.466667        0.400000       26.7
17.6
12      0.700000        97.933333        0.033333       16.4
13.3
13      4.333333        91.866667        0.166667       12.6
8.1
14      5.000000        87.666667        0.266667        7.4
4.5
15      5.266667        88.000000        1.600000       22.5
10.5
16      1.833333        96.500000        0.000000       82.3
38.5
17      4.166667        94.400000        0.233333       19.0
15.2
18      1.033333        98.933333        0.000000       84.2
47.4
19      7.333333        87.166667        0.700000        6.9
5.2
20      4.066667        91.900000        0.166667       46.6
17.3
21      6.733333        91.666667        0.133333       10.8
6.1
22     14.033333        80.533333        2.300000        5.0
2.3
23      7.000000        92.033333        0.700000        5.7
0.6
24      8.833333        88.466667        2.600000       14.0
12.0
25     11.533333        76.633333        0.200000        7.8
5.6
26      4.133333        94.333333        0.000000       59.0
```

```
39.4
27      8.833333         87.066667          0.166667        80.9
53.9
28      5.200000         90.366667          0.466667        12.4
6.4
29      4.566667         95.866667          0.800000        34.9
25.3
30      1.500000         96.566667          0.100000        15.5
0.0
31      5.600000         88.633333          0.000000        15.5
3.5
32      4.633333         93.266667          0.600000        12.7
4.3
33      5.100000         84.633333          0.933333         6.6
4.1
34      2.833333         92.766667          0.266667        13.5
9.1
35      8.866667         85.200000          4.833333        10.3
99.9

    Digital Library  ...  Girls' Toilet  Boys' Toilet  Electricity
Computers  \
0               2.2  ...      93.901747     90.864775    86.577934
9.983379
1               1.9  ...      99.519231     99.278846    92.788462
40.144231
2               5.0  ...      95.688319     83.597533    97.959579
10.361916
3               0.9  ...      68.692756     67.249514    53.788510
7.299473
4               0.5  ...      82.420020     76.498135    75.063672
4.355970
5               1.0  ...      97.457200     95.531584    87.547899
1.936349
6              10.3  ...      99.570815     98.712446   100.000000
82.403433
7               1.4  ...      96.128256     91.962769    91.400057
4.397296
8               4.9  ...      83.520199     84.997330   100.000000
46.058017
9               1.5  ...      99.205298     99.072848   100.000000
29.735099
10              2.5  ...      95.736384     94.091103    99.957290
20.309744
11              2.9  ...      95.330018     92.801146    97.820956
28.070471
12              1.3  ...      98.408032     97.581540    97.781229
14.732638
13              1.2  ...      82.374588     78.764103    72.942198
```

```
9.682347
14              1.9  ...      96.504292     94.430944    92.092297
5.415227
15              3.7  ...      97.251799     93.398300    98.587312
10.643558
16              7.4  ...      99.039409     97.678571    99.519704
83.177340
17              4.1  ...      89.877301     92.944785    92.229039
16.155419
18              7.9  ...     100.000000    100.000000   100.000000
78.947368
19              2.1  ...      94.784284     92.787979    74.844325
2.587951
20              4.8  ...      93.702842     90.935632    85.553579
23.425026
21              1.0  ...      75.265324     74.788824    54.494260
4.158545
22              0.7  ...      69.609589     73.068493    24.664384
1.890411
23              0.1  ...      86.832012     86.985426    79.621580
1.508566
24              1.3  ...      77.078734     76.195732    67.108168
10.743194
25              2.8  ...      90.126985     87.638664    76.651523
5.981602
26              4.2  ...      96.467391     95.516304   100.000000
54.483696
27              6.4  ...      98.631818     96.815999    99.978340
46.933324
28              2.1  ...      92.584584     87.983793    86.693992
9.531554
29              5.6  ...      92.136616     98.490866    98.411438
31.850675
30              0.0  ...      98.773830     98.005136   100.000000
9.889288
31              1.8  ...      77.589769     67.629924    90.337256
4.087459
32              0.7  ...      74.477582     72.448773    54.980726
9.373098
33              1.6  ...      96.734017     95.112263    81.322514
2.845528
34              1.5  ...      90.282709     88.042954    88.612755
14.345825
35              0.4  ...      98.810479     96.932787    96.157013
8.549354

      Internet  Drinking Water   Hand wash        Ramp  Ramp and
Handrails  \
0    33.912022       95.933021   93.643809   71.840993
```

```
49.720472
1    44.951923      100.000000  100.000000   62.740385
35.336538
2    56.085749       99.117001   98.422871   53.764448
35.918835
3    22.037191       67.943381   45.240078   24.535110
17.679711
4    11.709032       91.335710   89.212771   69.320889
61.565257
5    11.142596       99.260452   86.363978   71.034187
37.513015
6    98.712446      100.000000  100.000000   87.553648
54.077253
7    36.691322       98.313633   98.184456   83.481384
60.891492
8   100.000000      100.000000  100.000000  100.000000
100.000000
9    58.211921      100.000000  100.000000   60.728477
55.033113
10   91.961152       99.931292   96.729866   81.940911
59.302520
11   51.243362       99.490011   99.460507   72.646042
45.861081
12   34.474151       99.822498   99.389838   77.584868
65.492567
13   29.737893       95.306370   96.268009   39.347336
14.858532
14   37.289042       95.875599   91.450229   64.213577
37.075020
15   29.548725       98.037933   89.335513   70.417266
58.515370
16   95.190887       99.655172   99.217980   78.048030
60.929803
17   42.740286       91.717791   63.496933   78.732106
31.288344
18   97.368421      100.000000  100.000000   94.736842
76.315789
19   27.521460       94.148843   92.987052   80.285391
38.168687
20   47.947630       97.997354   98.443502   92.730259
78.846768
21   23.066927       98.353909   83.235867   49.620966
36.408924
22   16.849315       46.027397   37.883562   30.383562
21.226027
23    7.849655       90.207108   72.743544   44.285349
18.972130
24   50.883002       60.559235   60.522443   37.858720
16.409124
```

```
25   14.904240       97.010804   96.720232   80.473905
69.488369
26   98.369565      100.000000   99.592391   65.353261
57.880435
27   59.308328       99.974730   99.960290   89.704343
85.101621
28   59.859175       92.476474   96.789599   65.297585
41.439087
29   34.471803       99.285147   97.299444   29.070691
20.174742
30   37.560586      100.000000  100.000000   74.012347
41.092839
31   22.948727       86.393705   88.482696   75.187429
31.940673
32   18.178129       75.491986   85.595455   62.994522
35.280990
33   21.140536       96.585598   92.617049   64.452789
47.252901
34   27.372343       92.474249   96.419023   61.678720
40.368179
35   16.672296       99.490205   97.553407   77.247108
53.094655
```

```
    CWSN Toilet
0     26.961450
1     23.076923
2     19.913476
3     10.435748
4     18.229021
5     14.418505
6     73.390558
7     72.991577
8    100.000000
9      6.754967
10    34.933427
11    43.774762
12    23.507877
13     4.290922
14     5.194516
15    17.039895
16    25.548030
17    24.028630
18    60.526316
19    10.237136
20    52.686465
21     8.165475
22     2.616438
23    17.642547
24     5.445180
```

```
25    54.797643
26    38.043478
27    79.560305
28    19.833040
29    15.011914
30    29.895750
31     5.582248
32     8.784743
33    23.972114
34     6.890204
35    29.967069

[36 rows x 23 columns]
```

Exporting the finally modified dataset

```
#To export the final dataset
output_dir = 'output'
os.makedirs(output_dir, exist_ok=True)
final_set.to_excel(os.path.join(output_dir, 'modified_file.xlsx'),
index=False)
print(os.listdir(os.getcwd()))

['.virtual_documents', 'output']
```

Reviewing the final properties

```
# Reviewing the final dataset characteristics
final_set.info()
final_set.describe()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 23 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   India/ State /UT    36 non-null     object
 1   Total               36 non-null     int64
 2   Enrollment          36 non-null     int64
 3   Promotion Rate      36 non-null     float64
 4   Dropout Rate        36 non-null     float64
 5   Transition Rate     36 non-null     float64
 6   Repetition Rate     36 non-null     float64
 7   Projector           36 non-null     float64
 8   Smart Class         36 non-null     float64
 9   Digital Library     36 non-null     float64
 10  Library             36 non-null     float64
 11  Playground          36 non-null     float64
 12  Kitchen Garden      36 non-null     float64
```

```
 13  Girls' Toilet       36 non-null     float64
 14  Boys' Toilet        36 non-null     float64
 15  Electricity         36 non-null     float64
 16  Computers           36 non-null     float64
 17  Internet            36 non-null     float64
 18  Drinking Water      36 non-null     float64
 19  Hand wash           36 non-null     float64
 20  Ramp                36 non-null     float64
 21  Ramp and Handrails  36 non-null     float64
 22  CWSN Toilet         36 non-null     float64
dtypes: float64(20), int64(2), object(1)
memory usage: 6.6+ KB

              Total    Enrollment  Promotion Rate  Dropout Rate  \
count  3.600000e+01  3.600000e+01       36.000000     36.000000
mean   8.271583e+04  1.473162e+07       94.027778      5.426852
std    2.466222e+05  4.403938e+07        3.739217      3.292317
min    3.800000e+01  1.358600e+04       83.600000      0.000000
25%    3.834000e+03  4.214425e+05       92.233333      2.958333
50%    2.571350e+04  6.013938e+06       94.533333      5.050000
75%    6.203375e+04  1.167980e+07       96.608333      7.408333
max    1.489115e+06  2.652358e+08      100.000000     14.033333

       Transition Rate  Repetition Rate  Projector  Smart Class  \
count        36.000000        36.000000  36.000000    36.000000
mean         90.790741         0.634259  26.763889    18.291667
std           6.647953         0.954338  24.781008    20.452948
min          74.066667         0.000000   3.200000     0.000000
25%          87.141667         0.125000  10.050000     5.050000
50%          91.283333         0.266667  15.950000     9.900000
75%          96.016667         0.708333  35.800000    22.750000
max         107.666667         4.833333  85.000000    99.900000

       Digital Library     Library  ...  Girls' Toilet  Boys'
Toilet  \
count        36.000000   36.000000  ...      36.000000     36.000000

mean          2.766667   86.179096  ...      90.958729     89.134328

std           2.419445   19.535682  ...       9.202318      9.652707

min           0.000000   23.753425  ...      68.692756     67.249514

25%           1.150000   81.700304  ...      86.004059     84.647381

50%           1.900000   94.735123  ...      95.057151     92.794563

75%           4.125000   98.213139  ...      97.694908     96.845196

max          10.300000  100.000000  ...     100.000000    100.000000
```

```
       Electricity  Computers    Internet  Drinking Water   Hand wash
\
count    36.000000  36.000000   36.000000       36.000000   36.000000
mean     85.985777  20.722066   43.553107       93.283795   90.090622
std      17.272819  23.184282   28.028206       12.115272   15.391794
min      24.664384   1.508566    7.849655       46.027397   37.883562
25%      78.879066   5.160744   22.720843       92.475918   89.030252
50%      91.746177  10.172648   35.582737       98.175783   96.725049
75%      98.820410  28.486628   56.617292       99.849697   99.407505
max     100.000000  83.177340  100.000000      100.000000  100.000000

              Ramp  Ramp and Handrails  CWSN Toilet
count   36.000000           36.000000    36.000000
mean    67.313985           46.958913    28.170788
std     18.746960           20.354230    24.278010
min     24.535110           14.858532     2.616438
25%     61.441159           35.322651     9.874038
50%     70.725726           43.650084    21.495199
75%     79.120428           59.699763    35.710940
max    100.000000          100.000000   100.000000

[8 rows x 22 columns]
```

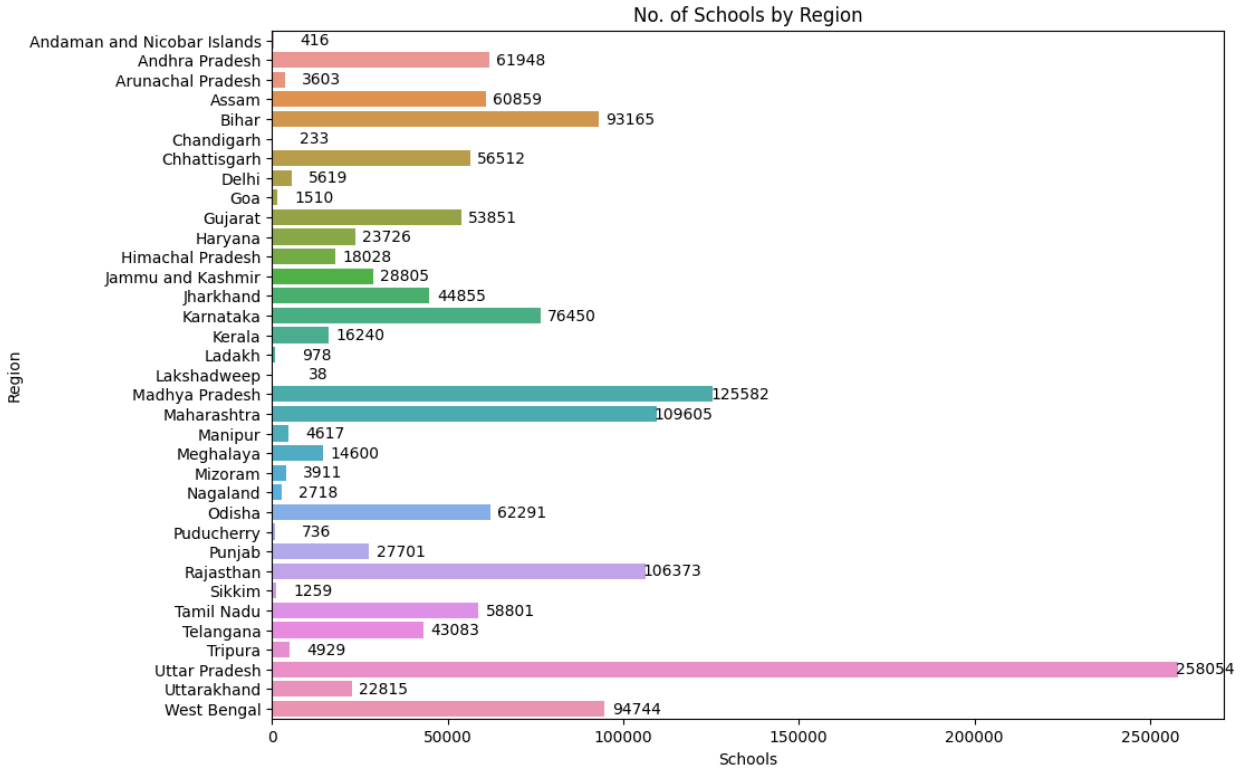
Exploratory Data Analysis

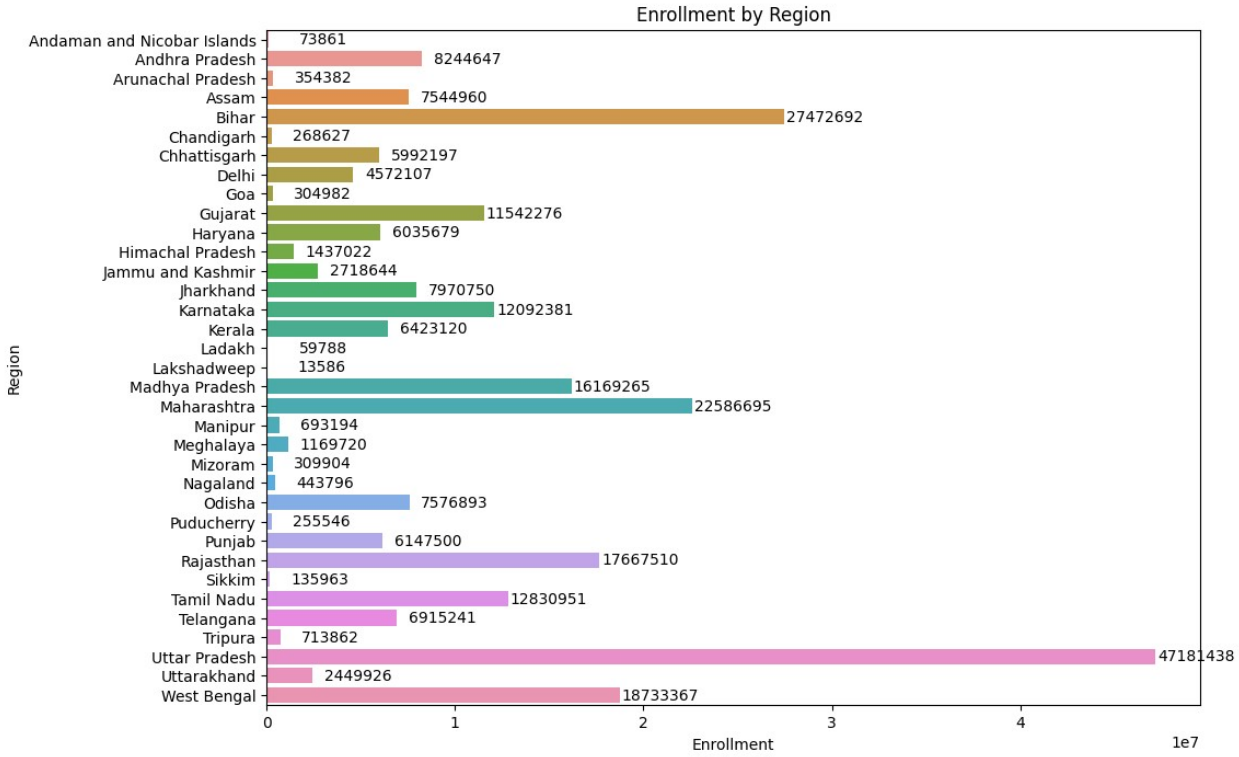
Visualize regional disparities and temporal trends using charts and maps. Perform descriptive statistical analysis to understand the distribution and variability of infrastructural and technological elements.

Quantitative Analysis of Schools and Enrollment

```
# Bar Plot: No. of Schools by Region
plt.figure(figsize=(11, 8))
sns.barplot(x='Total', y='India/ State /UT',
data=final_set.drop(index=0))
plt.title('No. of Schools by Region')
plt.xlabel('Schools')
plt.ylabel('Region')
```

```
for index, value in enumerate(final_set.drop(index=0)['Total']):
    plt.text(value+16000, index, f'{value}', ha='right', va='center',
color='black', fontsize=10)
plt.show()

# Bar Plot: Total Enrollment by Region
plt.figure(figsize=(11, 8))
sns.barplot(x='Enrollment', y='India/ State /UT',
data=final_set.drop(index=0))
plt.title('Enrollment by Region')
plt.xlabel('Enrollment')
plt.ylabel('Region')
for index, value in enumerate(final_set.drop(index=0)['Enrollment']):
    plt.text(value+4200000, index, f'{value}', ha='right',
va='center', color='black', fontsize=10)
plt.show()
```





Summary of Schools and Enrollment by Indian States and UTs

The dataset provides information on the total number of schools and the corresponding enrollment figures across various states and Union Territories (UTs) in India. Here's a brief analysis:

Overall Summary:

India has a total of 1,489,115 schools with a combined enrollment of 265,235,830 students. Uttar Pradesh has the highest number of schools, totaling 258,054, with an enrollment of 47,181,438 students. Madhya Pradesh follows with 125,582 schools and an enrollment of 16,169,265 students. Maharashtra has 109,605 schools and 22,586,695 students enrolled. Lakshadweep has the smallest number of schools at just 38, with an enrollment of 13,586 students. Sikkim and Chandigarh also have relatively few schools, with 1,259 and 233 schools respectively.

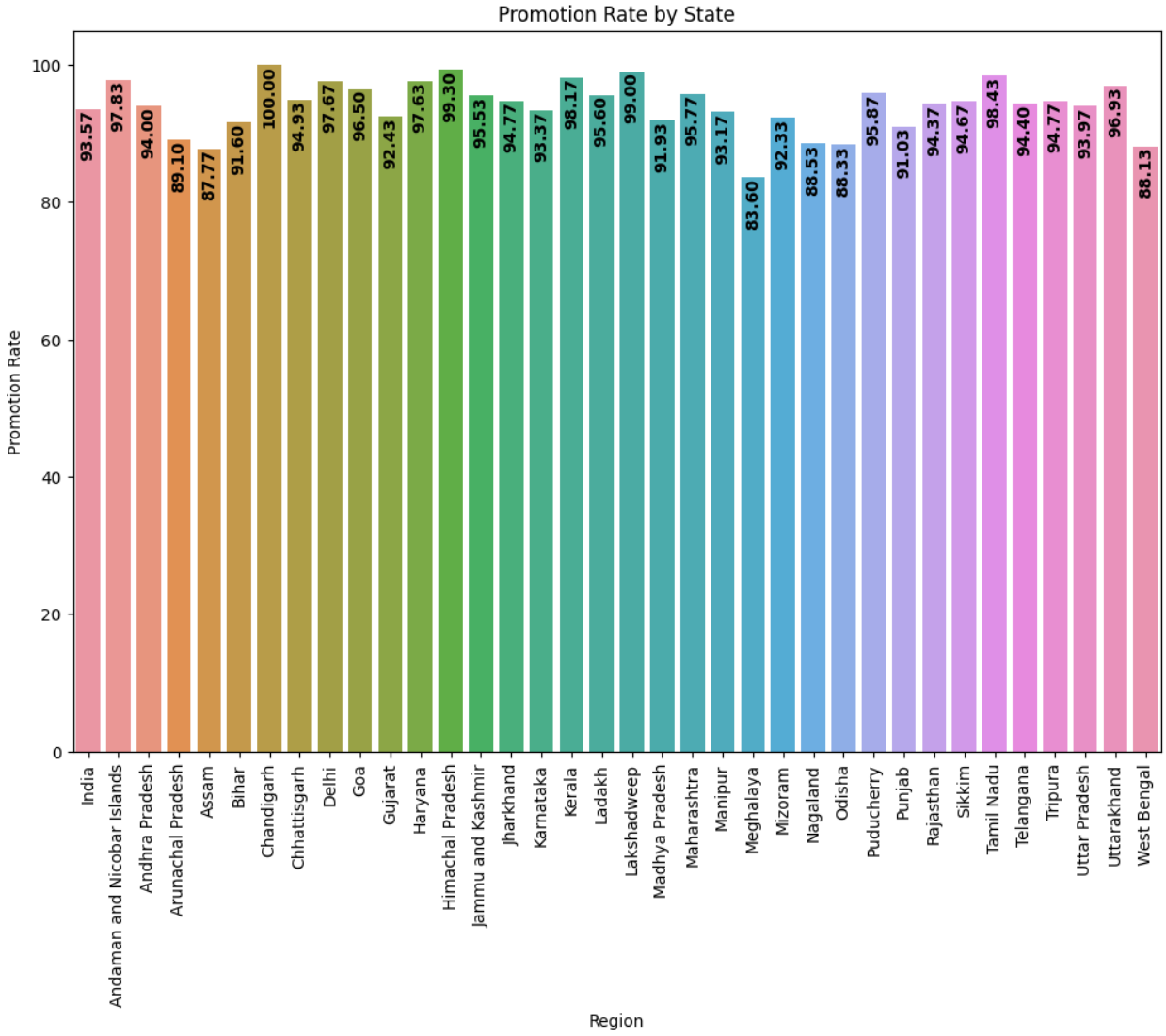
There is a clear disparity in the number of schools and enrollment figures across different states, influenced by regional population size, educational policies, and infrastructure development. States with fewer schools but high enrollment may indicate higher student-to-school ratios, which could reflect on the quality and accessibility of education.

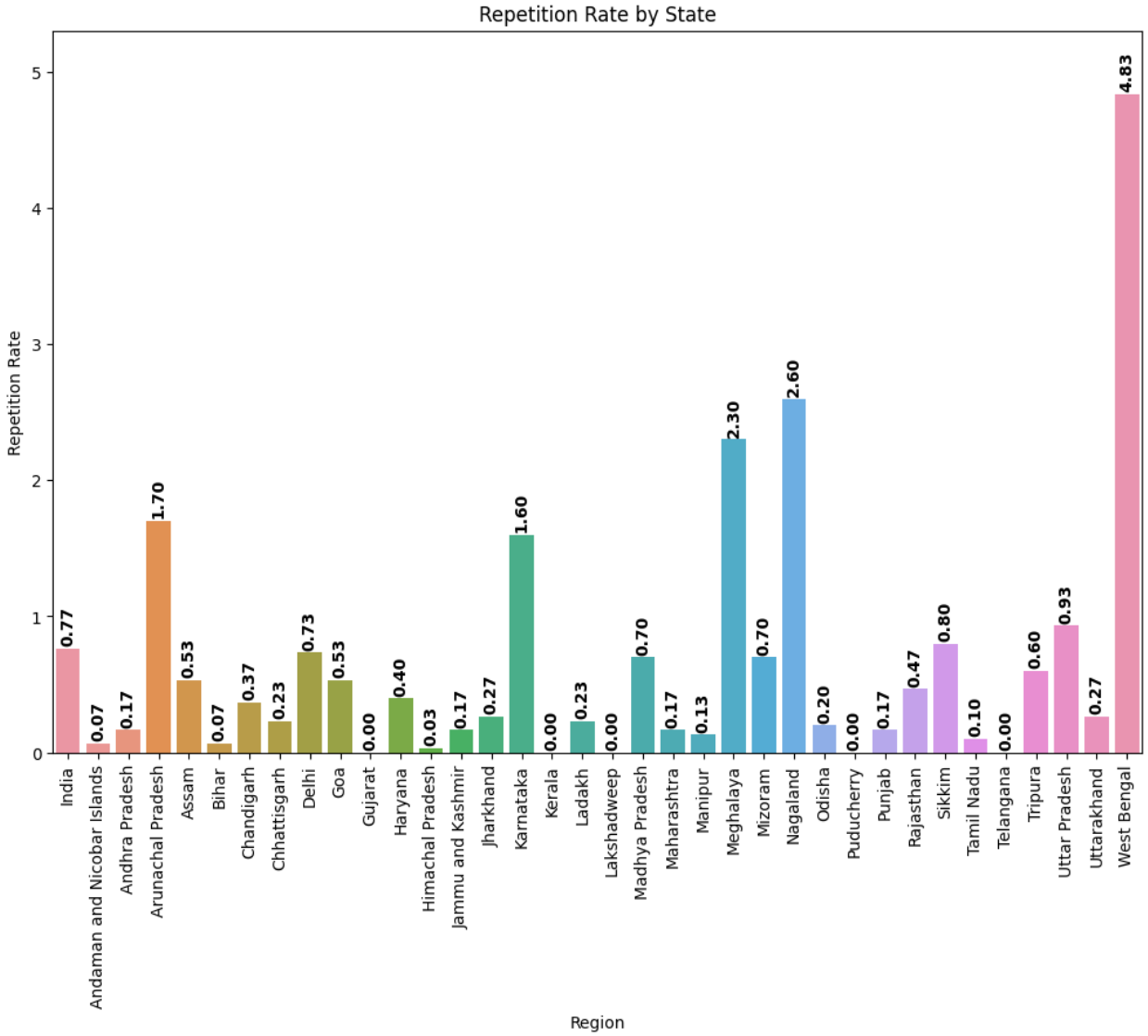
Analysing the Education Metrics across India

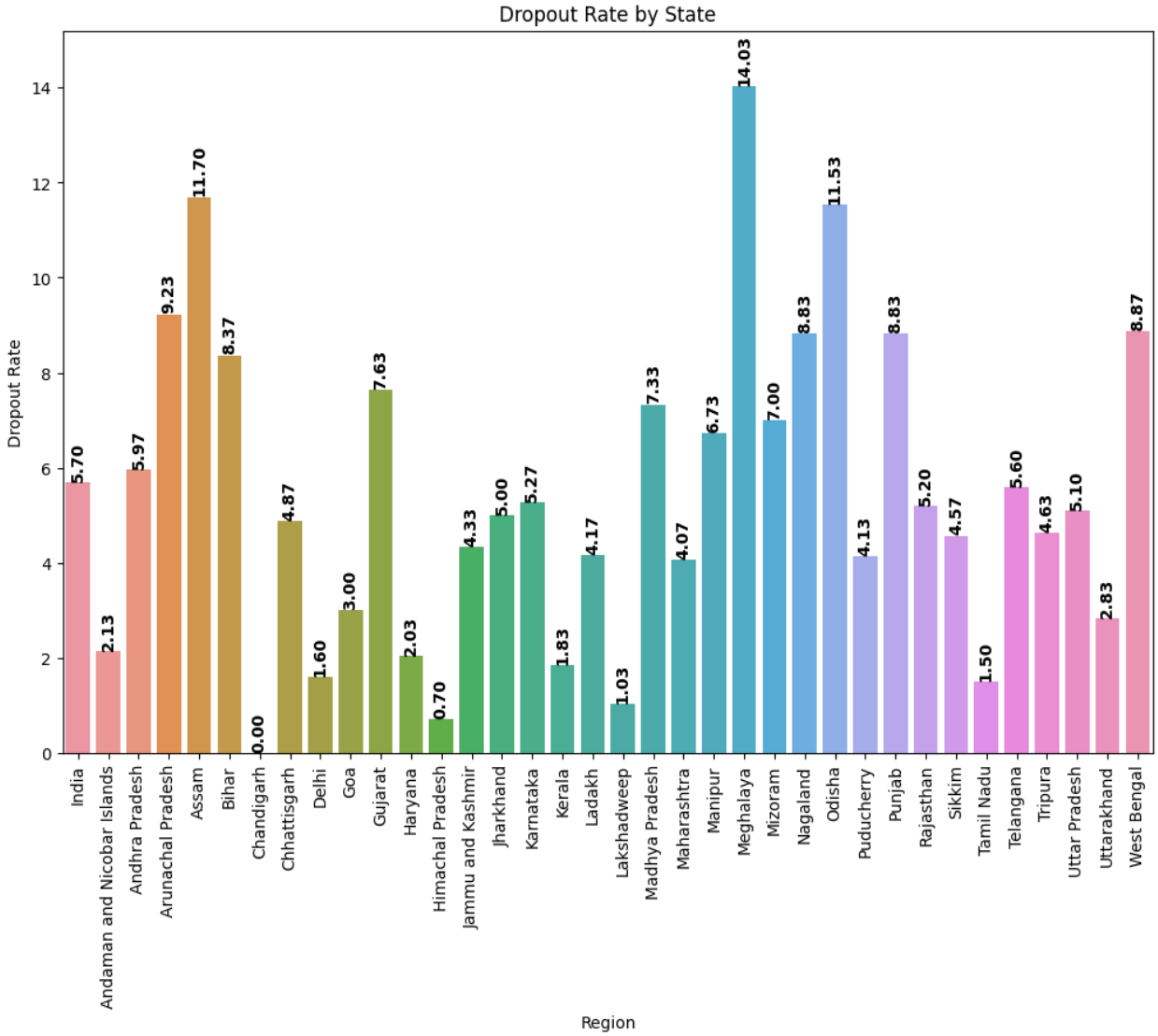
```
# Bar Plot: Promotion by Region
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='India/ State /UT', y='Promotion Rate',
data=final_set)
plt.title('Promotion Rate by State')
```

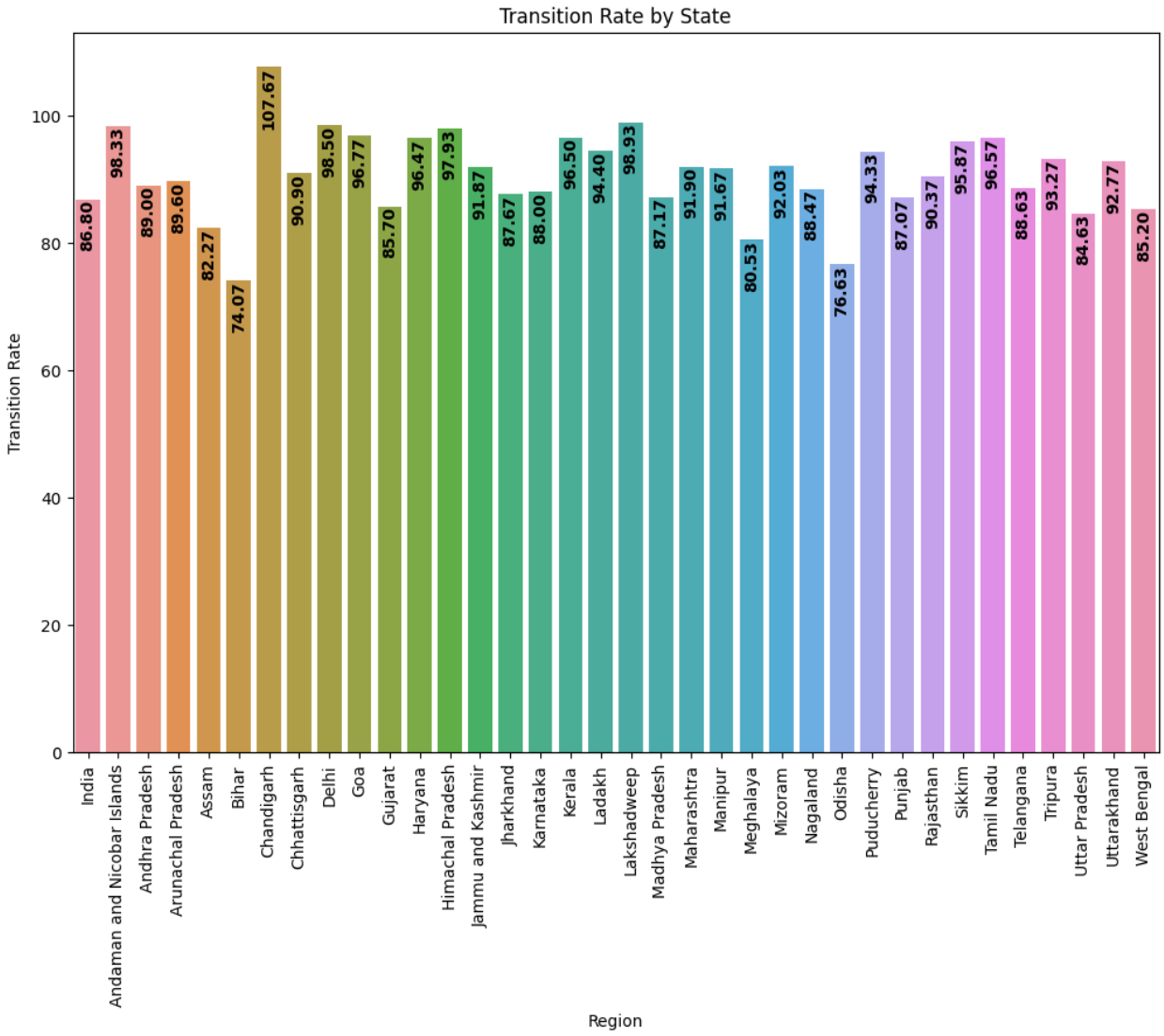
```
plt.xlabel('Region')
plt.ylabel('Promotion Rate')
plt.xticks(rotation=90)
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='top',
                fontsize=10, color='black', weight='bold',rotation=90)
plt.show()

# Bar Plot: Repetition by Region
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='India/ State /UT', y='Repetition Rate',
data=final_set)
plt.title('Repetition Rate by State')
plt.xlabel('Region')
plt.ylabel('Repetition Rate')
plt.xticks(rotation=90)
ax.set_ylim(0, 5.3)
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()+0.02),

                ha='center', va='bottom',
                fontsize=10, color='black', weight='bold',rotation=90)
plt.show()

# Bar Plot: Dropout by Region
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='India/ State /UT', y='Dropout Rate',
data=final_set)
plt.title('Dropout Rate by State')
plt.xlabel('Region')
plt.ylabel('Dropout Rate')
plt.xticks(rotation=90)
ax.set_ylim(0, 15.2)
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()+0.02),

                ha='center', va='bottom',
                fontsize=10, color='black', weight='bold',rotation=90)
plt.show()

# Bar Plot: Transition by Region
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='India/ State /UT', y='Transition Rate',
data=final_set)
plt.title('Transition Rate by State')
plt.xlabel('Region')
```

```
plt.ylabel('Transition Rate')
plt.xticks(rotation=90)
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='top',
                fontsize=10, color='black', weight='bold',rotation=90)

plt.show()
```




Repetition Rate by State
Repetition Rate
Region
0
1
2
3
4
5
0.77
0.07
0.17
1.70
0.53
0.07
0.37
0.23
0.73
0.53
0.00
0.40
0.03
0.17
0.27
1.60
0.00
0.23
0.00
0.70
0.17
0.13
2.30
0.70
2.60
0.20
0.00
0.17
0.47
0.80
0.10
0.00
0.60
0.93
0.27
4.83
India
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Ladakh
Lakshadweep
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal


Dropout Rate by State
Dropout Rate
Region
0
2
4
6
8
10
12
14
India
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Ladakh
Lakshadweep
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
5.70
2.13
5.97
9.23
11.70
8.37
0.00
4.87
1.60
3.00
7.63
2.03
0.70
4.33
5.00
5.27
1.83
4.17
1.03
7.33
4.07
6.73
14.03
7.00
8.83
11.53
4.13
8.83
5.20
4.57
1.50
5.60
4.63
5.10
2.83
8.87



Summary of Educational Metrics by Indian States and UTs

The dataset provides key educational metrics for various states and Union Territories (UTs) in India, including the Promotion Rate, Dropout Rate, Transition Rate, and Repetition Rate. Here's a comprehensive summary:

Overall Metrics:

1. Promotion Rate: Average of 93.57%, reflecting the percentage of students advancing to the next grade.
2. Dropout Rate: Average of 5.7%, indicating the percentage of students who leave school before completion.
3. Transition Rate: Average of 86.8%, showing the percentage of students transitioning to the next educational level.
4. Repetition Rate: Average of 0.77%, representing the percentage of students repeating a grade.

Promotion Rate Insights:

- Chandigarh has a perfect promotion rate of 100%.
- Kerala and Tamil Nadu also show high promotion rates of 98.17% and 98.43%, respectively.
- Himachal Pradesh and Lakshadweep have promotion rates of 99.3% and 99%.
- Meghalaya has the lowest promotion rate at 83.6%.
- West Bengal and Odisha follow with 88.13% and 88.33%, respectively.

Dropout Rate Insights:

- Meghalaya has the highest dropout rate at 14.03%.
- Assam and West Bengal also report high dropout rates of 11.7% and 8.87%, respectively.
- Himachal Pradesh and Chandigarh have very low dropout rates, 0.7% and 0%.

Transition Rate Insights:

- Chandigarh has an exceptionally high transition rate of 107.67%, which might indicate additional factors like data reporting or cross-state transitions.
- Kerala and Tamil Nadu also have high transition rates of 96.5% and 96.57%, respectively.
- Assam and West Bengal have lower transition rates of 82.27% and 85.2%, respectively.

Repetition Rate Insights:

- West Bengal has the highest repetition rate at 4.83%.
- Nagaland and Meghalaya also show relatively high repetition rates of 2.6% and 2.3%, respectively.
- Tamil Nadu and Himachal Pradesh have very low repetition rates of 0.1% and 0.03%, respectively.

Regional Trends:

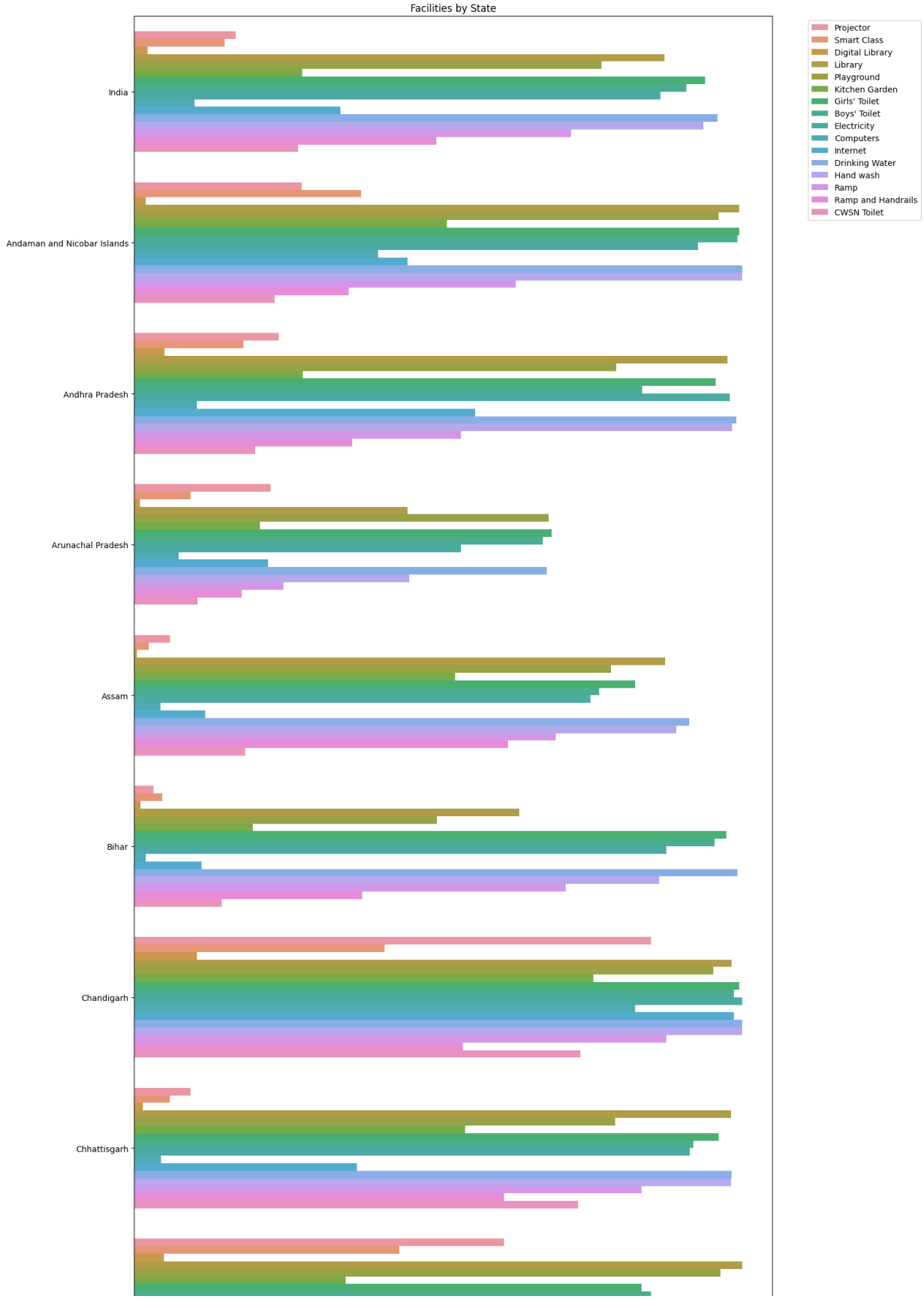
- Southern India states such as Kerala, Tamil Nadu, and Goa generally show high promotion rates and low dropout rates, indicating strong educational performance.
- Northern and Northeastern India states like Meghalaya and Nagaland report higher dropout and repetition rates, highlighting potential areas for improvement.
- Central India states like Madhya Pradesh and Uttar Pradesh have moderate metrics, reflecting a balance between performance and challenges.

Disparities and Opportunities:

- The data shows significant variability in educational outcomes across states, with some regions excelling in promotion rates while others struggle with higher dropout and repetition rates.
- States with lower dropout rates and higher promotion rates, such as Chandigarh and Kerala, can serve as models for educational success.
- States with higher dropout and repetition rates may benefit from targeted interventions and educational reforms to improve student retention and success.

Analysing the Facilities Coverage across India

```
# Stacked Bar Graph: Facilities Coverage by different states and UTs
facilities_coverage =
final_set.drop(final_set.columns[[1,2,3,4,5,6]],axis=1).melt(id_vars='
India/ State /UT', var_name='Facility', value_name='Coverage')

plt.figure(figsize=(14, 120))
ax = sns.barplot(x='Coverage', y='India/ State /UT', hue='Facility',
data=facilities_coverage,dodge=True)
plt.title('Facilities by State')
plt.xlabel('Coverage')
plt.ylabel('Facility')
plt.xticks(rotation=90)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


Facilities by State
India
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Projector
Smart Class
Digital Library
Library
Playground
Kitchen Garden
Girls' Toilet
Boys' Toilet
Electricity
Computers
Internet
Drinking Water
Hand wash
Ramp
Ramp and Handrails
CWSN Toilet

Summary of Facility Availability in Indian Schools

The dataset provides information on the availability of various facilities in schools across Indian states and Union Territories (UTs). The metrics include the percentage of schools equipped with facilities such as *Projectors, Smart Classes, Digital Libraries, Libraries, Playgrounds, Kitchen Gardens, Girls' Toilets, Boys' Toilets, Electricity, Computers, Internet, Drinking Water, Hand Wash Stations, Ramps, Ramps and Handrails, and CWSN (Children with Special Needs) Toilets.*

Basic Facility Availability:

- Drinking Water is widely available with an average of 95.9%, reflecting good access to basic hydration needs.
- Electricity is present in an average of 86.6% of schools, which is critical for modern education.
- Girls' Toilets and Boys' Toilets have high availability, 93.9% and 90.9% respectively, indicating a strong focus on basic sanitation.
- Hand Wash Stations are available in 93.6% of schools, promoting hygiene practices among students.
- Libraries are highly available across states, with an average availability of 87.3%. Many states like Chandigarh, Delhi, and Goa have near-total availability.

Variability in Advanced Facilities:

- Smart Classes and Projectors show significant variability, with Delhi and Chandigarh leading in these advanced facilities, whereas states like Bihar and Jharkhand have much lower percentages.
- Digital Libraries are relatively rare, with an average availability of only 2.2%, indicating a potential area for development.
- Computers are available in 33.9% of schools, and Internet access is slightly higher at 33.9%. This indicates room for improvement in integrating technology into the educational environment.

Playground and Kitchen Gardens:

- Playgrounds are available in 77% of schools on average, which is relatively high but varies significantly between states.
- Kitchen Gardens are present in 27.6% of schools, showing a moderate commitment to promoting nutrition and environmental education.

Accessibility Features:

- Ramps are present in 49.7% of schools, and Ramps and Handrails are available in 71.8%. These features are crucial for ensuring accessibility for children with disabilities.
- On average, 49.7% of Indian schools have CWSN (Children with Special Needs) toilets. Delhi and Chandigarh lead in availability, while Assam and Bihar lag behind, highlighting the need for improved accessibility in many areas.

State-Specific Highlights:

- Chandigarh stands out with high availability across most categories, particularly in Projectors, Smart Classes, Libraries, and Digital Libraries.

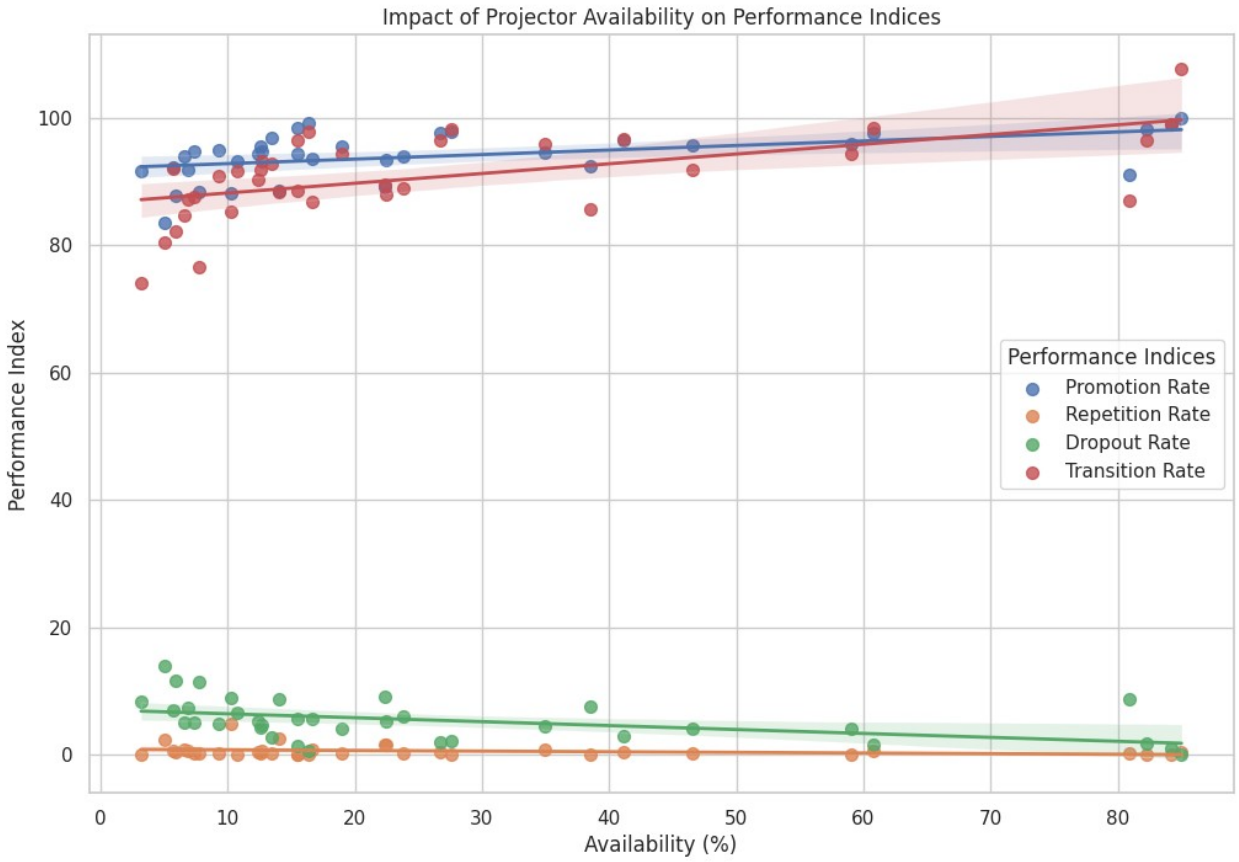
- West Bengal shows exceptionally high availability of Playgrounds (99.9%) but less availability of Smart Classes and Digital Libraries.
- Madhya Pradesh and Uttar Pradesh have lower availability of advanced facilities but maintain good basic infrastructure.
- States like Assam, Bihar, and Meghalaya have lower percentages in advanced facilities such as Smart Classes and Projectors but maintain a decent level of basic amenities.

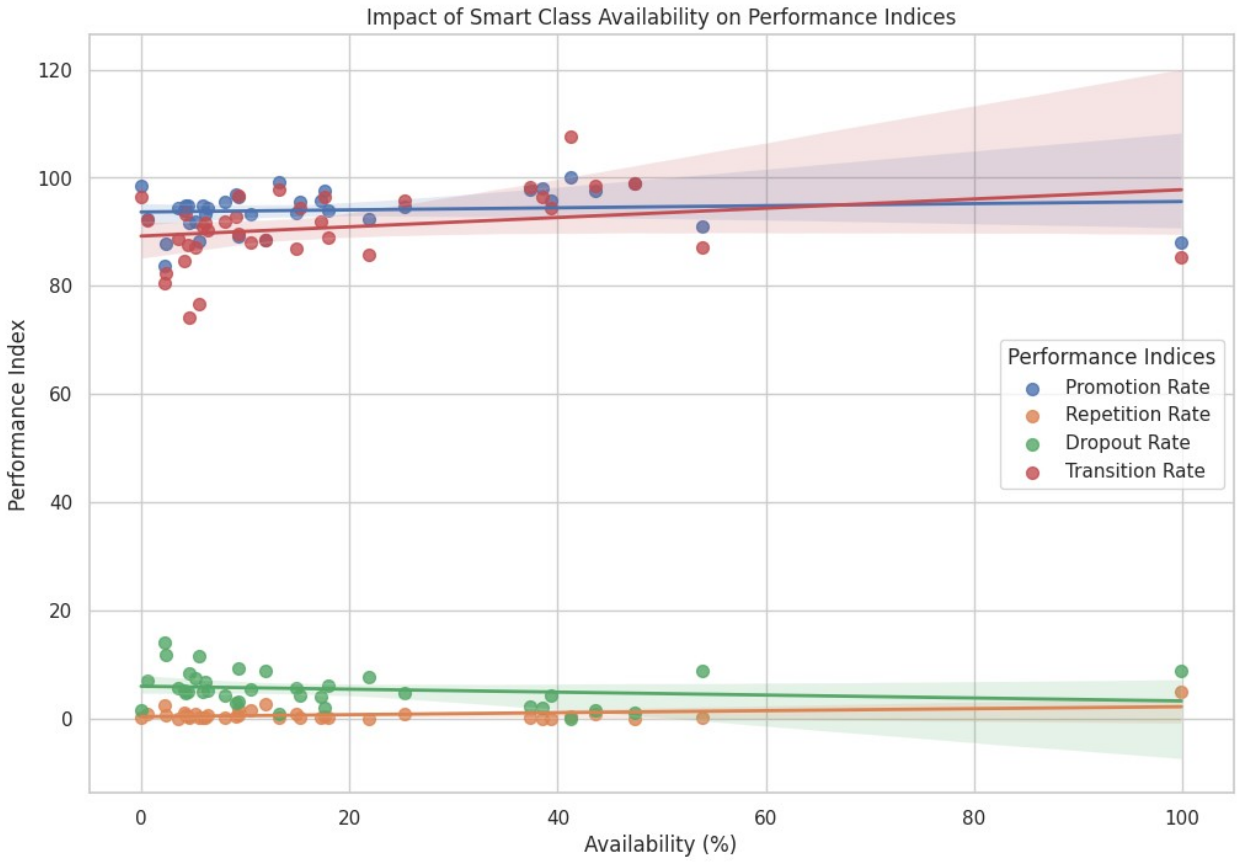
Conclusions:

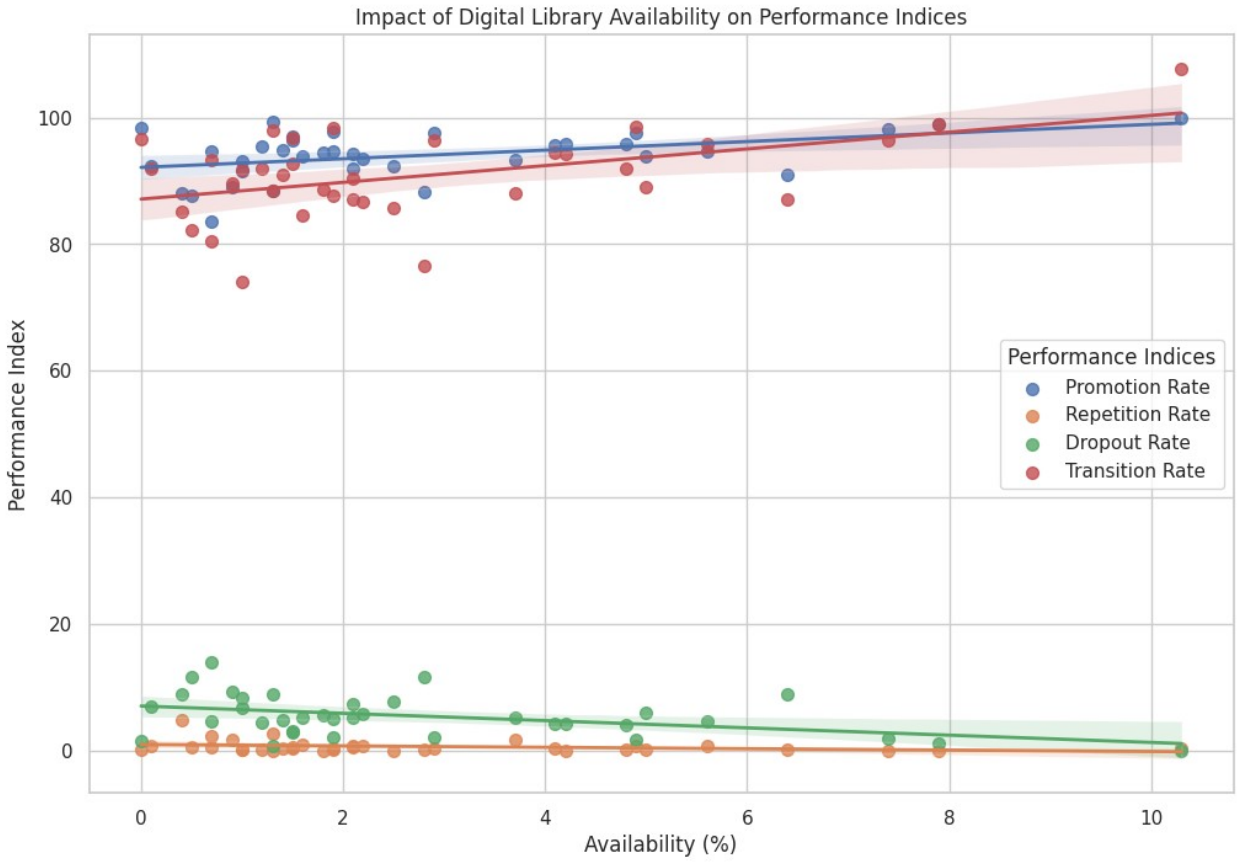
- There is a considerable disparity in the availability of educational facilities across different states and UTs.
- While basic facilities such as libraries, drinking water, and sanitation are broadly available, advanced facilities like smart classes and digital libraries are less common, suggesting areas where infrastructure could be enhanced.
- States like Delhi, Chandigarh, and Kerala show higher levels of facility availability and could serve as models for improving educational infrastructure.

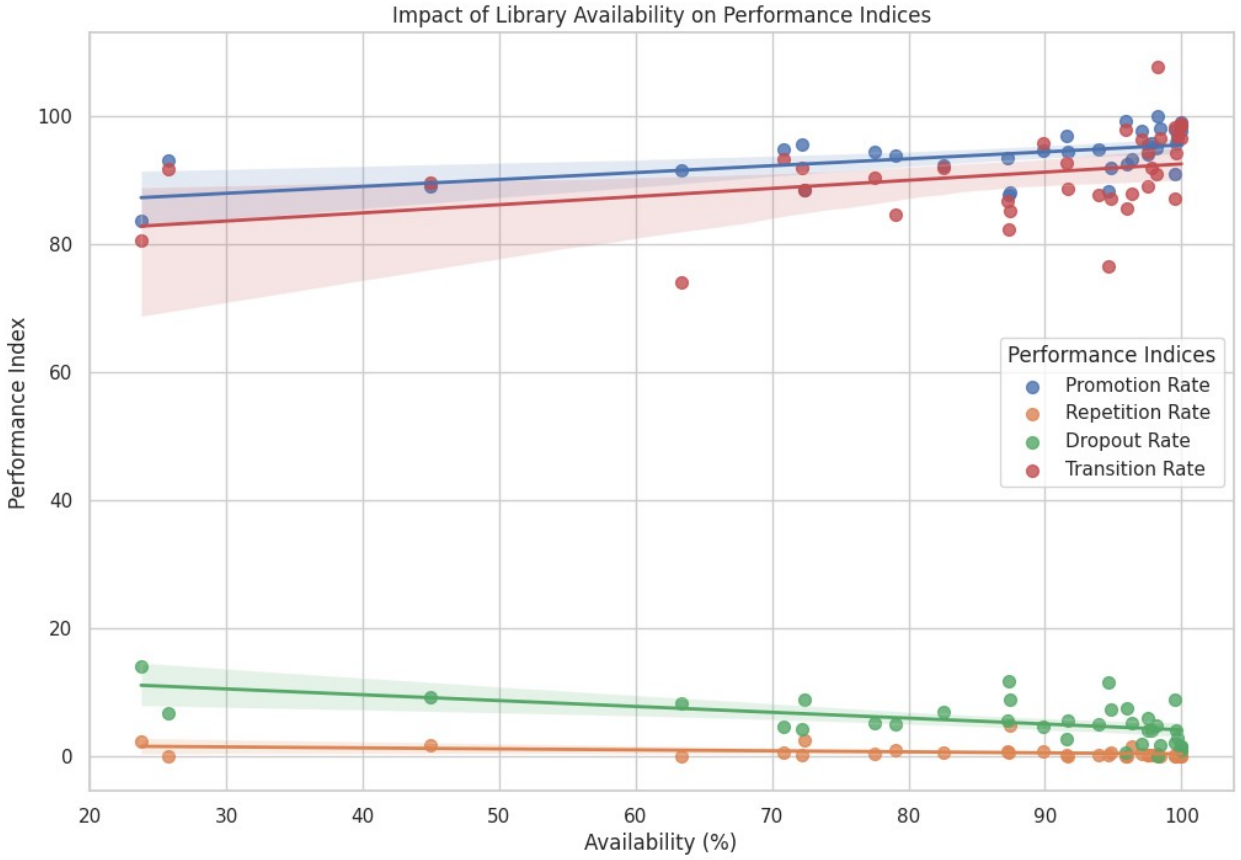
Analysing the impact of each facility

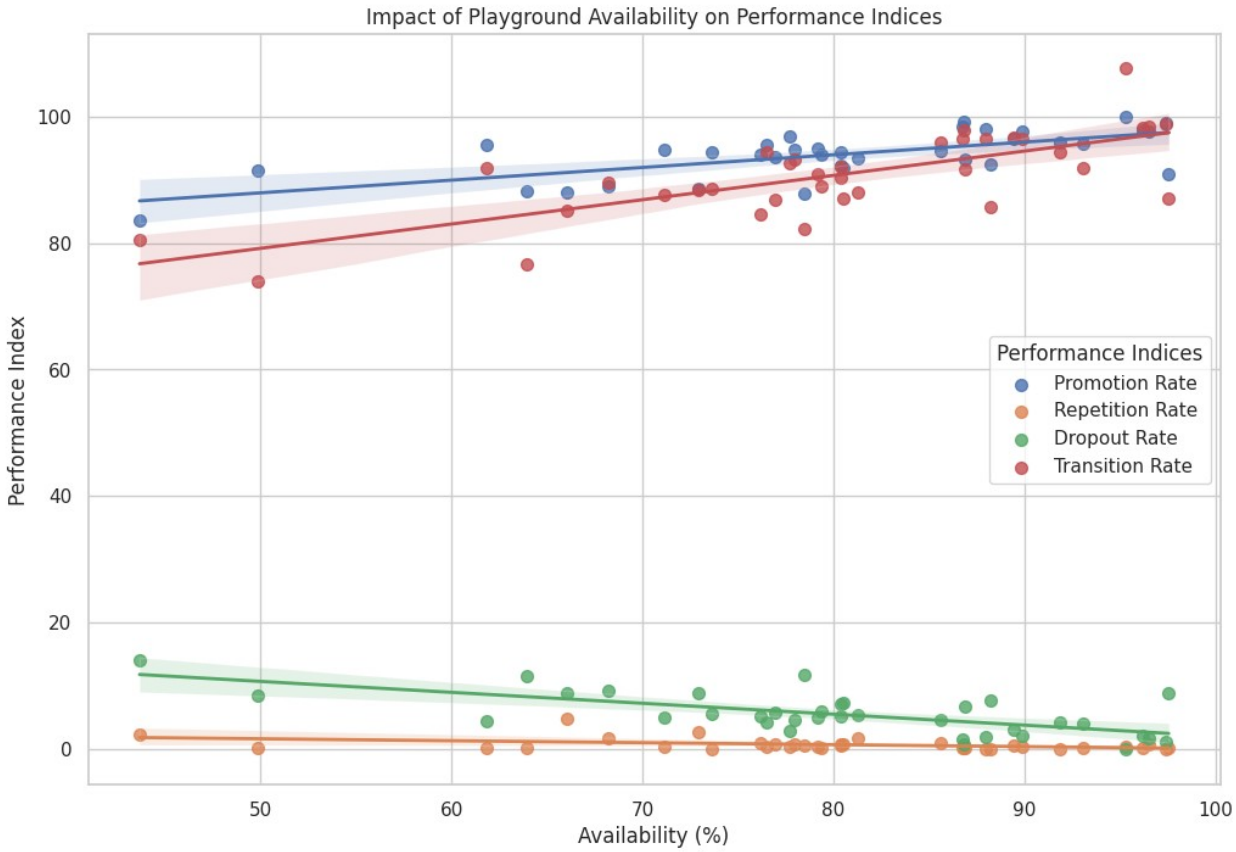
```
# Scatter Plot with Regression Line: Impact of Facility Availability
on Performance Indices
sns.set(style="whitegrid")
performance_indices = ['Promotion Rate', 'Repetition Rate', 'Dropout
Rate', 'Transition Rate']
facilities=['Projector','Smart Class','Digital
Library','Library','Playground','Kitchen Garden","Girls'
Toilet","Boys' Toilet","Electricity",'Computers','Internet','Drinking
Water','Hand wash','Ramp','Ramp and Handrails','CWSN Toilet']
for facility in facilities:
    plt.figure(figsize=(12, 8))
    for index in performance_indices:
        sns.regplot(x=facility, y=index, data=final_set, label=index,
scatter_kws={'s':50}, line_kws={'linewidth':2})
    plt.title(f'Impact of {facility} Availability on Performance
Indices')
    plt.xlabel('Availability (%)')
    plt.ylabel('Performance Index')
    plt.legend(title='Performance Indices')
    plt.show()
```

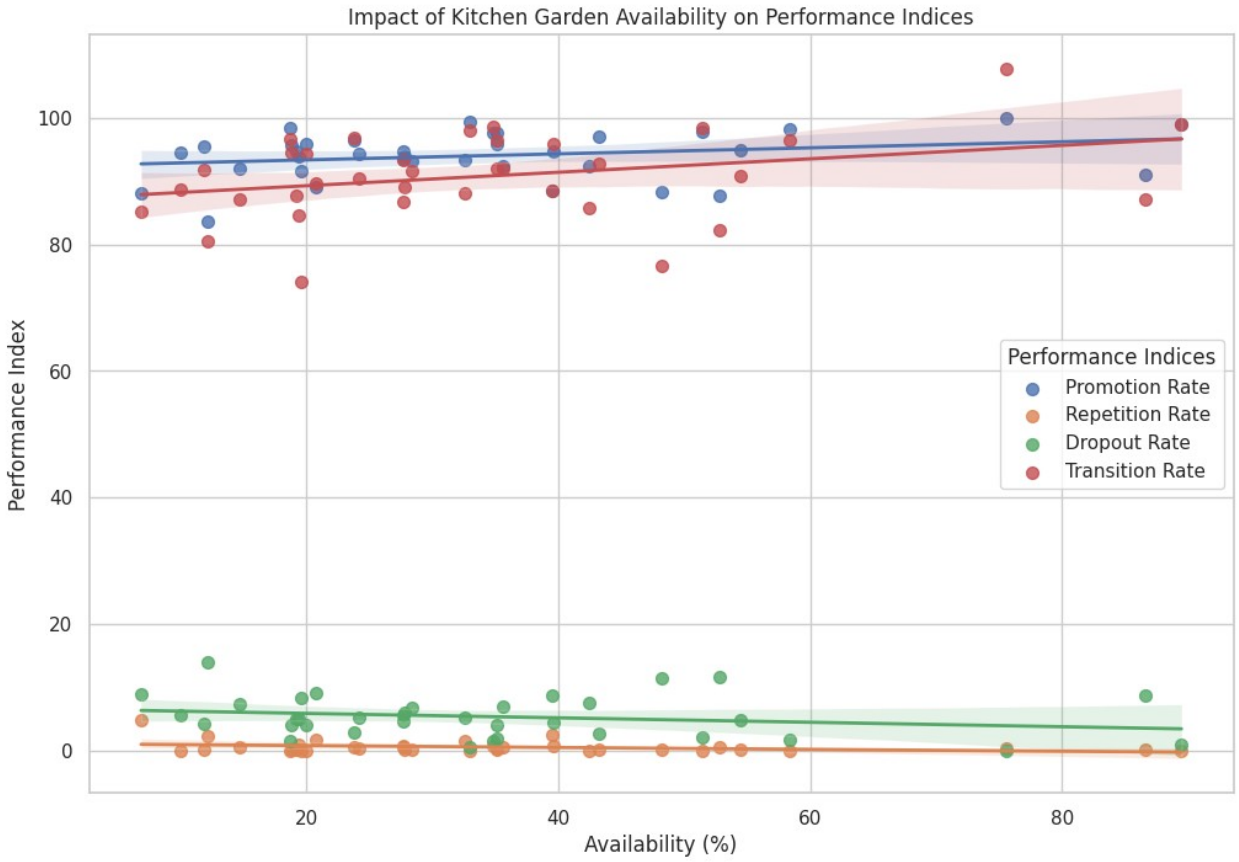

Impact of Projector Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80

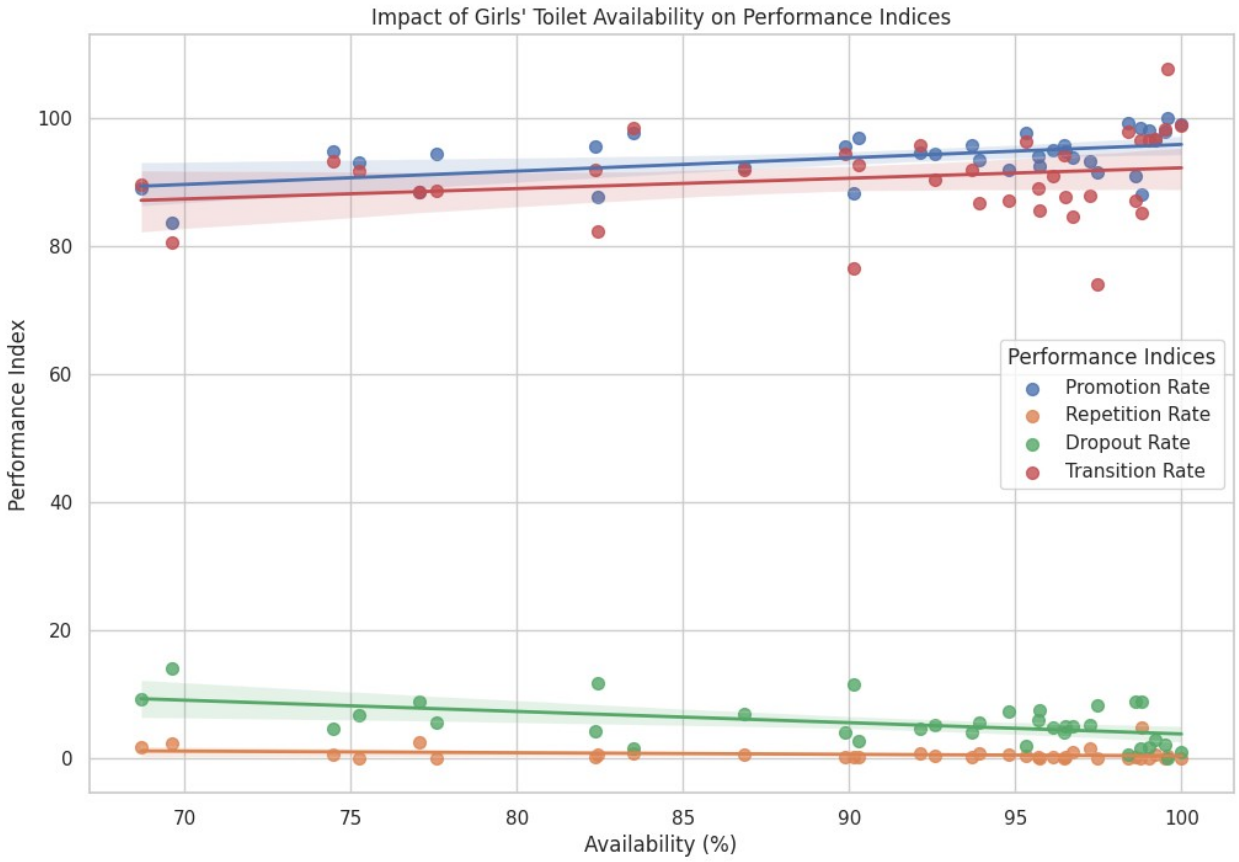

Impact of Smart Class Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
120
100
80
60
40
20
0
0
20
40
60
80
100

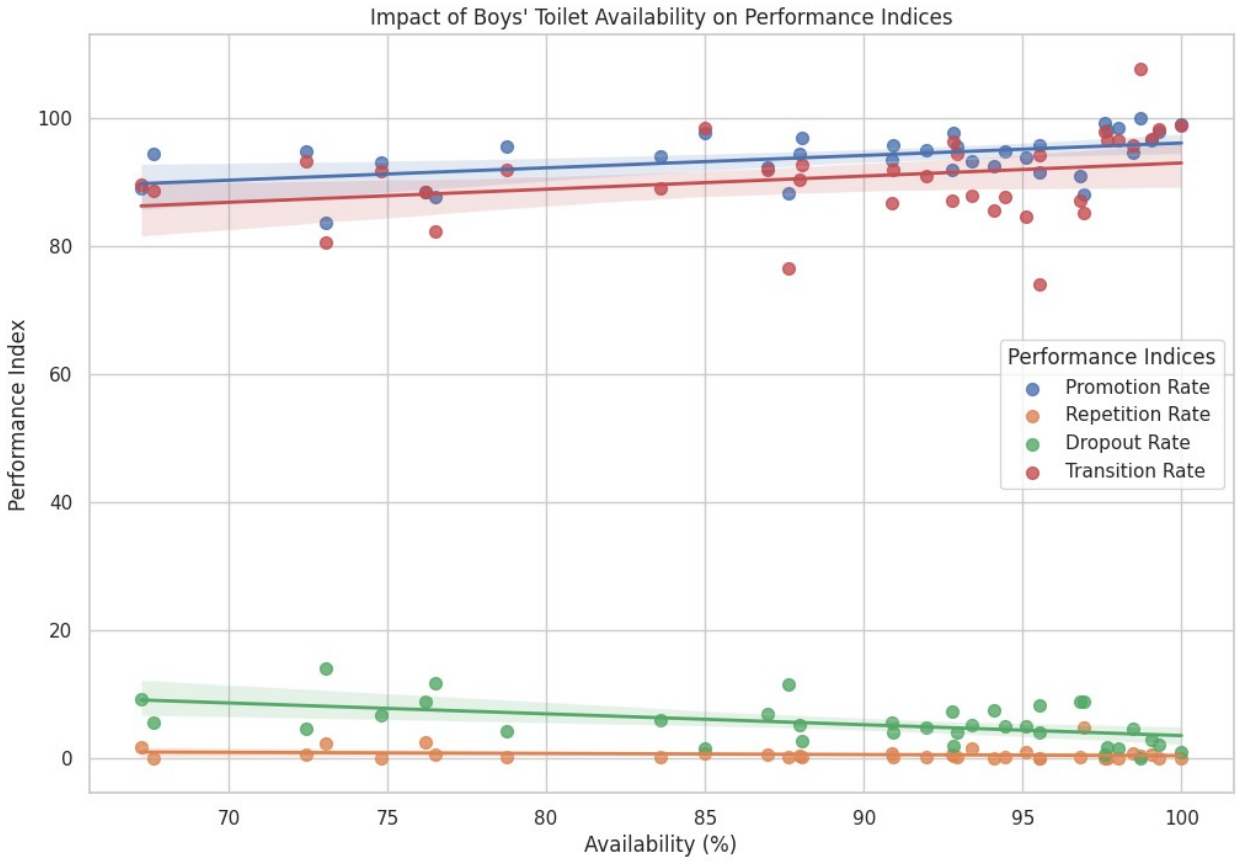

Impact of Digital Library Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
0
2
4
6
8
10

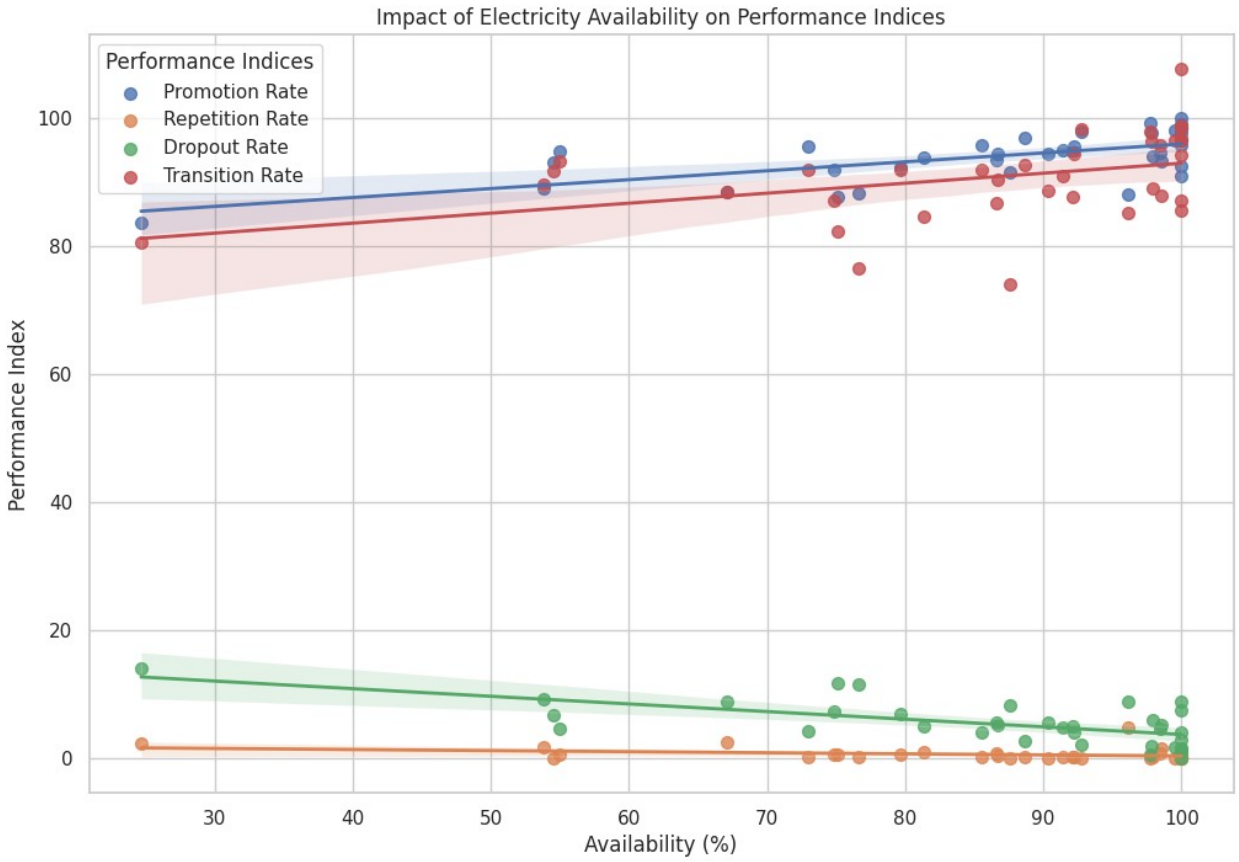

Impact of Library Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
20
30
40
50
60
70
80
90
100

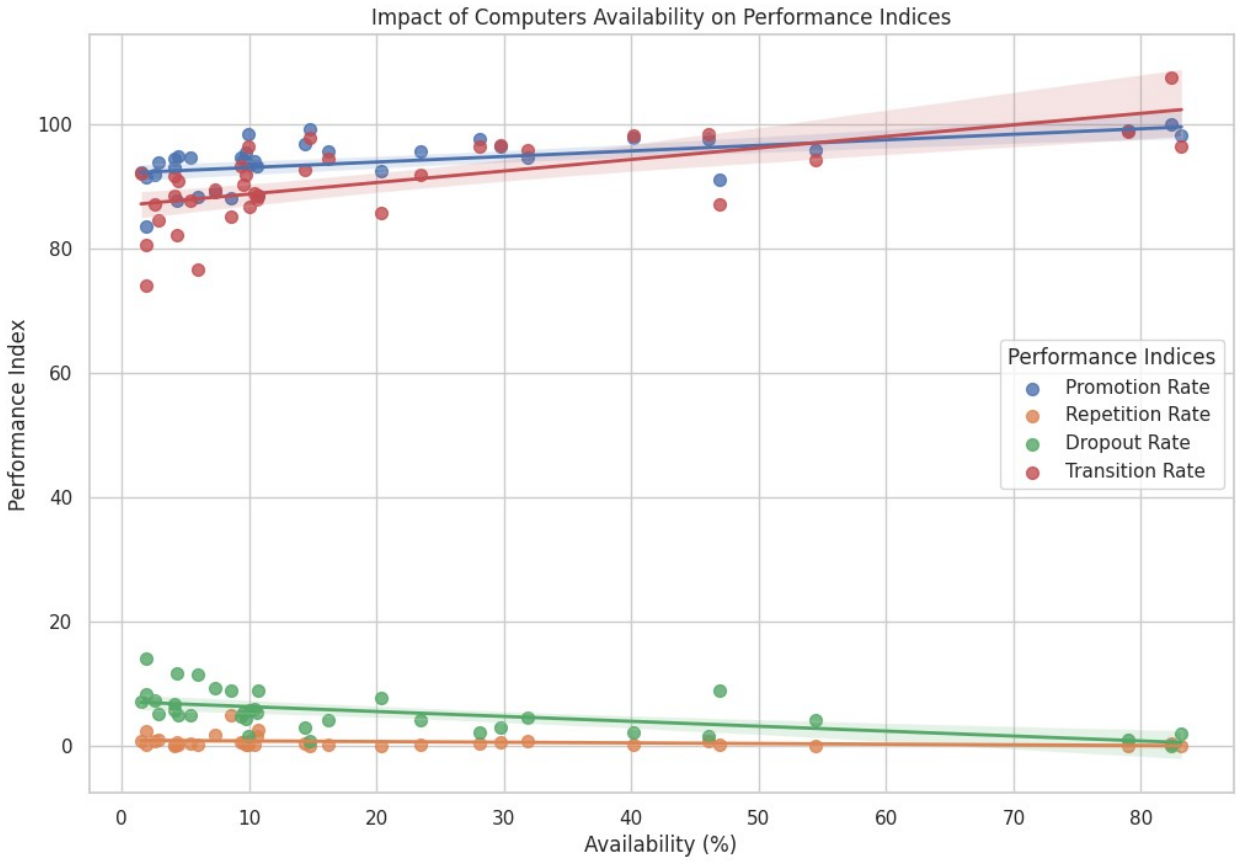

Impact of Playground Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
50
60
70
80
90
100

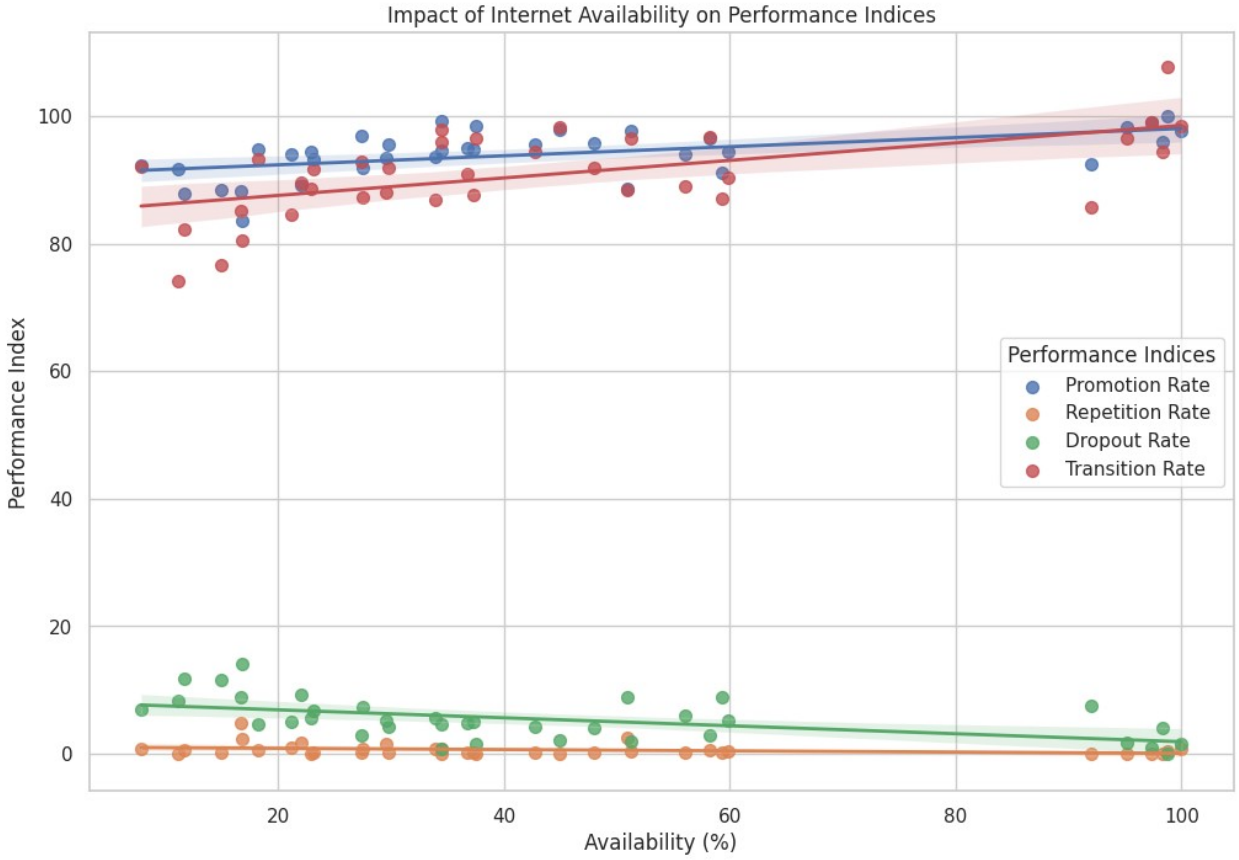

Impact of Kitchen Garden Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
20
40
60
80

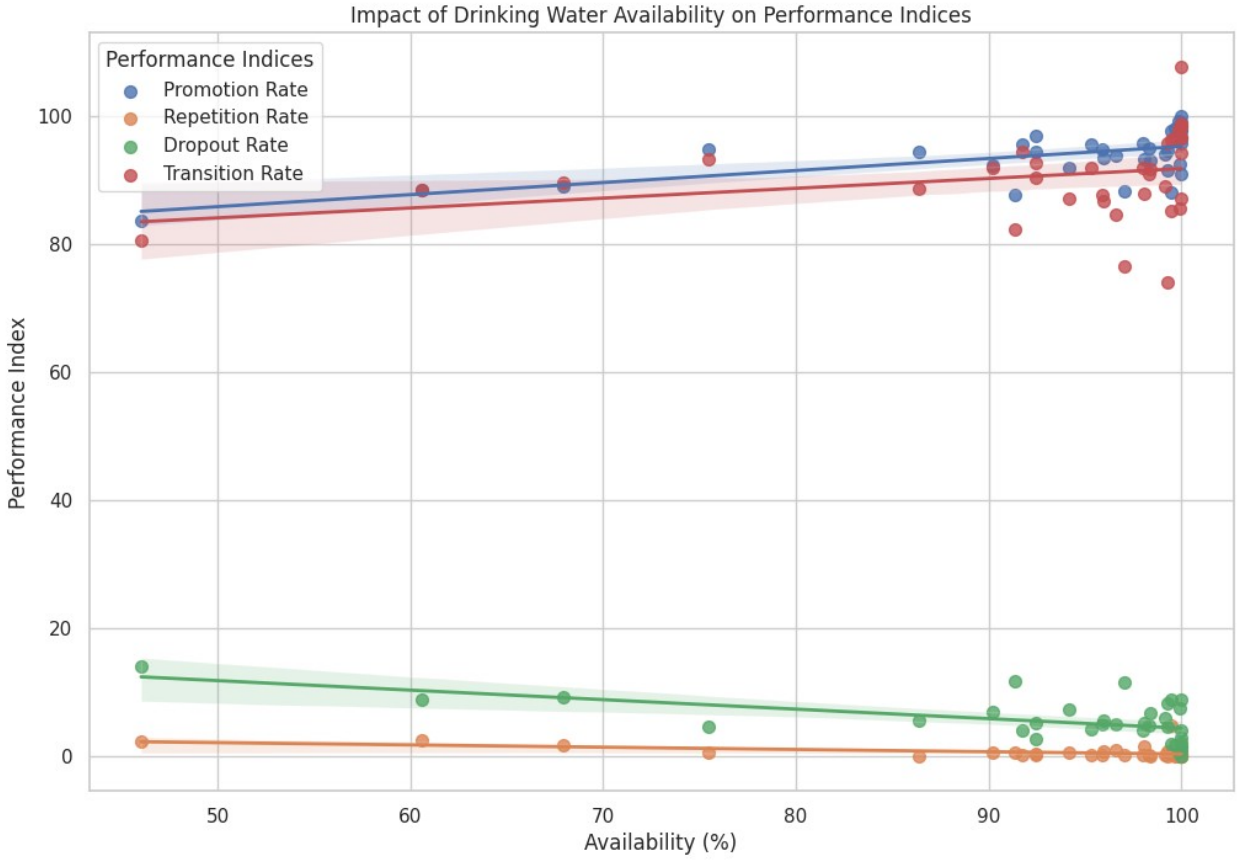

Impact of Girls' Toilet Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
70
75
80
85
90
95
100

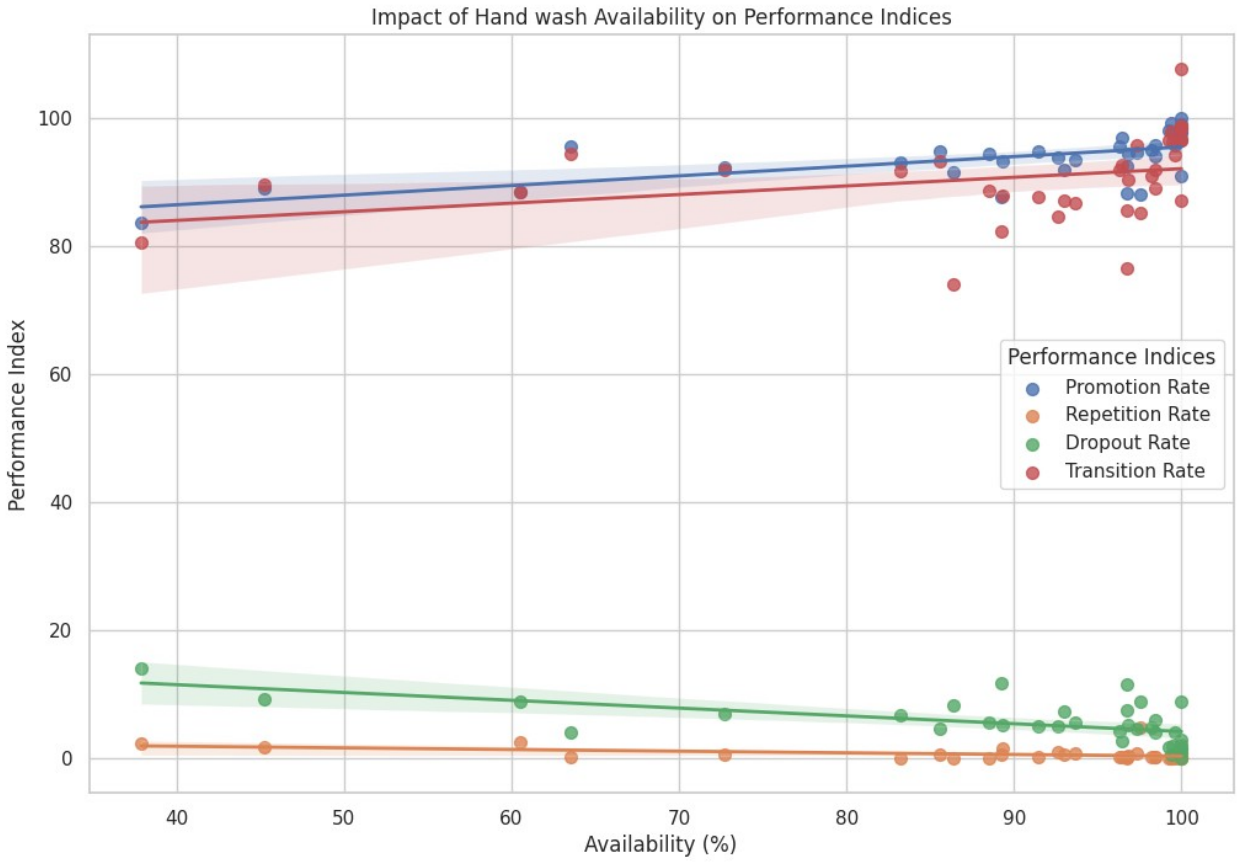

Impact of Boys' Toilet Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
70
75
80
85
90
95
100

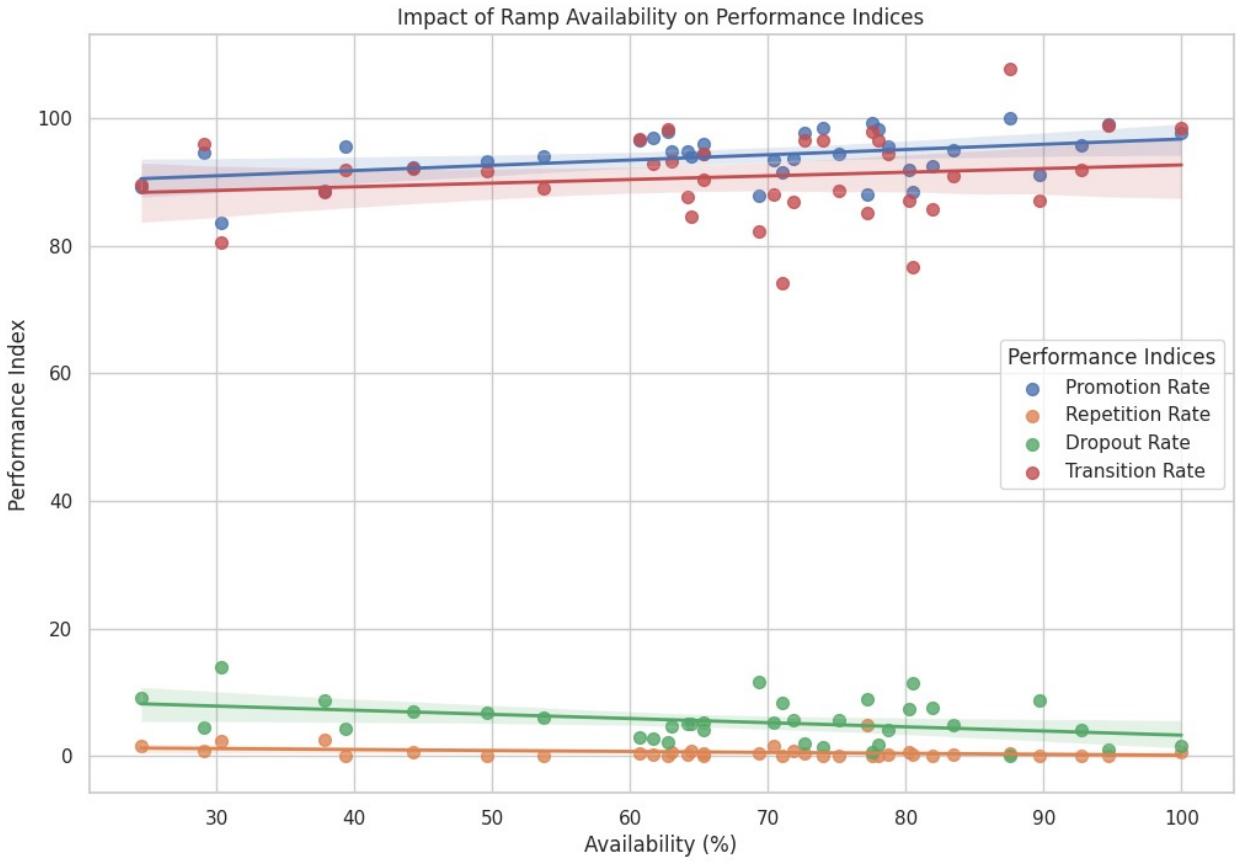

Impact of Electricity Availability on Performance Indices
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
Performance Index
Availability (%)
0
20
40
60
80
100
30
40
50
60
70
80
90
100

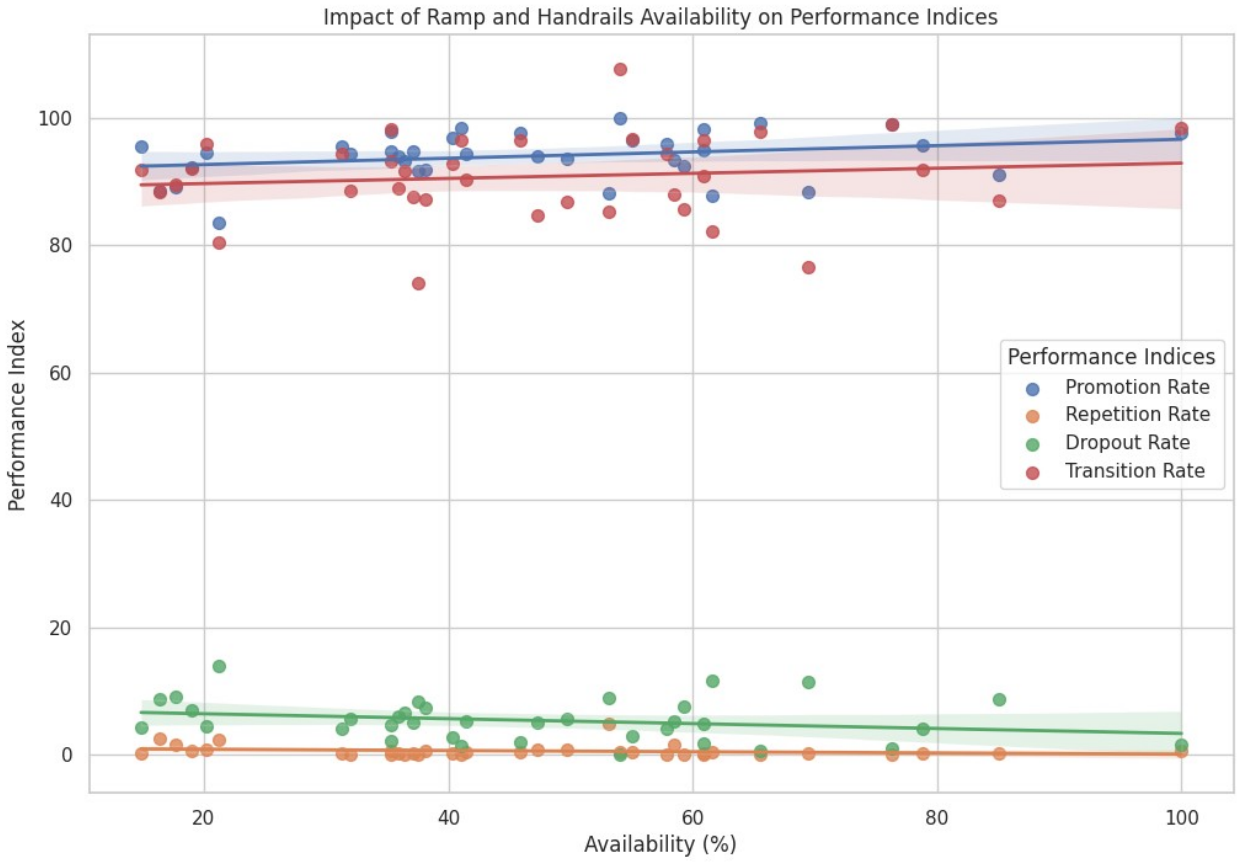

Impact of Computers Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80

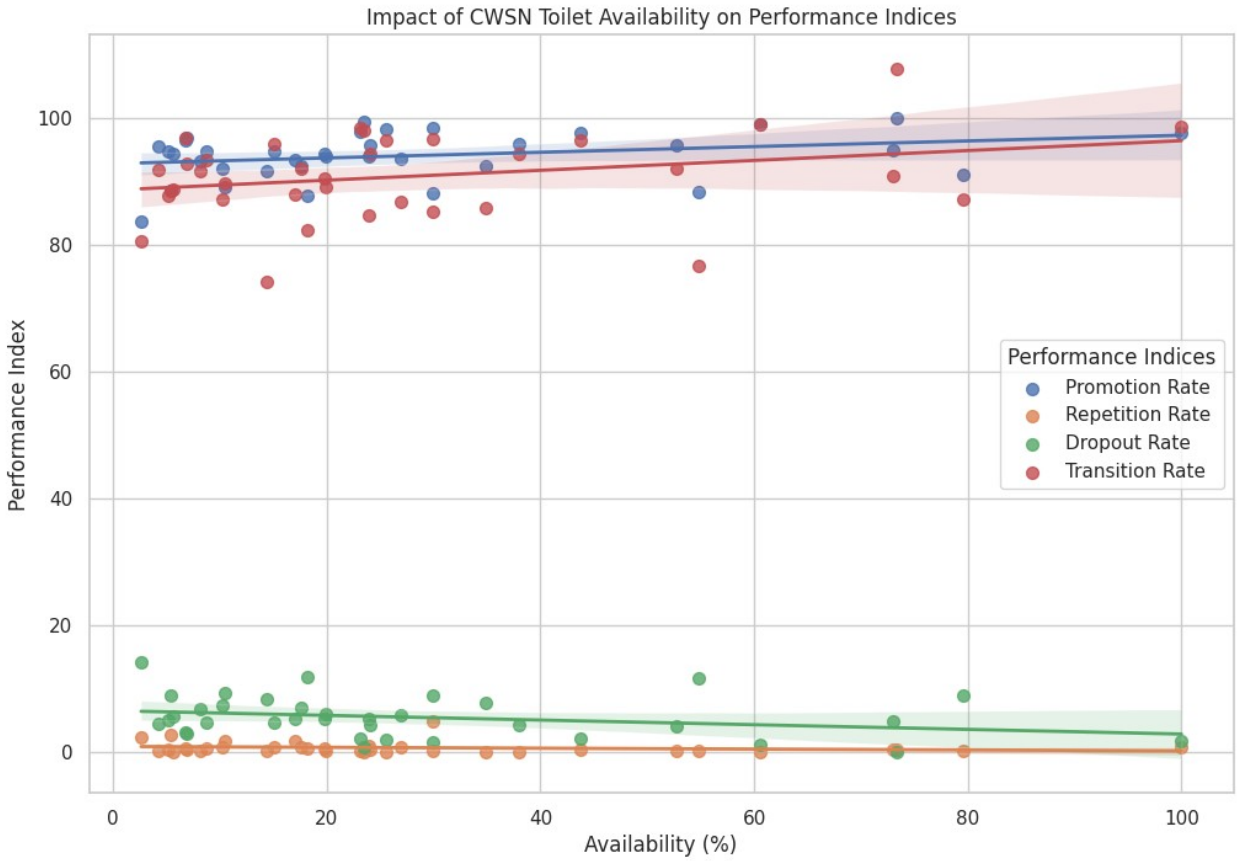

Impact of Internet Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
20
40
60
80
100


Impact of Drinking Water Availability on Performance Indices
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
Performance Index
Availability (%)
0
20
40
60
80
100
50
60
70
80
90
100


Impact of Hand wash Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
40
50
60
70
80
90
100


Impact of Ramp Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100
30
40
50
60
70
80
90
100


Impact of Ramp and Handrails Availability on Performance Indices
Performance Index
Availability (%)
Performance Indices
Promotion Rate
Repetition Rate
Dropout Rate
Transition Rate
0
20
40
60
80
100



Impact of Overall Facility Availability on Educational Performance Indices

Increased availability of educational(Infrastructural as well as Technological) facilities generally correlates with improved performance metrics across various indices:

1. **Promotion Rate**: Schools with more comprehensive facilities typically achieve higher promotion rates, indicating better student progression through grades.
2. **Dropout Rate**: Higher facility availability is often associated with lower dropout rates, suggesting that well-equipped schools enhance student retention.
3. **Transition Rate**: Schools with a greater range of facilities tend to have higher transition rates, reflecting smoother student advancement between educational stages.
4. **Repetition Rate**: Enhanced facility availability generally correlates with lower repetition rates, signifying fewer students repeating grades.

Analysis and Recommendations on Current Scenario of Infratech Facilities

1. Projector and Smart Classrooms

Current Statistics:

- Projectors: Ranges from 3.2% in Bihar to 85% in Chandigarh.
- Smart Classes: Ranges from 0% in Tamil Nadu to 99.9% in West Bengal.

Impact on Quality Education:

Projectors and smart classes enhance interactive learning and can significantly improve engagement and comprehension. High availability in states like Chandigarh and West Bengal suggests better technology adoption and potential for improved student outcomes.

Suggestions for Improvement:

- Increase Adoption: Invest in technology infrastructure in states with low percentages (e.g., Bihar, Tamil Nadu).
- Training: Provide training for teachers on effective use of projectors and smart classes to maximize benefits.

2. Digital Libraries and Libraries

Current Statistics:

- Digital Libraries: Ranges from 0.1% in Mizoram to 10.3% in Chandigarh.
- Libraries: Ranges from 23.75% in Meghalaya to 100% in Delhi and Lakshadweep.

Impact on Quality Education:

Digital libraries and physical libraries provide essential resources for research, reading, and self-learning, which support academic achievement and foster independent study habits.

Suggestions for Improvement:

- Expand Resources: Increase investment in digital and physical library resources in states with low coverage.
- Promote Usage: Implement programs to encourage students to utilize library resources effectively.

3. Playground and Kitchen Gardens

Current Statistics:

- Playgrounds: Ranges from 44.99% in Arunachal Pradesh to 100% in Delhi and Lakshadweep.
- Kitchen Gardens: Ranges from 6.85% in West Bengal to 75.54% in Chandigarh.

Impact on Quality Education:

Playgrounds support physical development and social skills. Kitchen gardens offer experiential learning opportunities and promote healthy eating habits.

Suggestions for Improvement:

- Enhance Facilities: Develop or improve playgrounds and kitchen gardens, especially in states with low availability (e.g., Arunachal Pradesh, West Bengal).
- Integrate Learning: Use kitchen gardens as part of the curriculum to teach students about agriculture and nutrition.

4. Toilets (Girls', Boys', and CWSN)

Current Statistics:

- Girls' Toilets: Ranges from 49.72% in India to 100% in several states.
- Boys' Toilets: Ranges from 37.51% in Bihar to 100% in several states.
- CWSN Toilets: Ranges from 16.41% in Nagaland to 100% in states like Delhi and Lakshadweep.

Impact on Quality Education:

Adequate sanitation facilities are critical for maintaining student health and attendance. CWSN toilets are essential for inclusive education for children with special needs.

Suggestions for Improvement:

- Upgrade Facilities: Ensure all schools have sufficient and functional sanitation facilities.
- Focus on Inclusivity: Prioritize the installation and maintenance of CWSN toilets to support children with special needs.

5. Electricity, Computers, and Internet

Current Statistics:

- Electricity: Ranges from 74.8% in Madhya Pradesh to 100% in several states.
- Computers: Ranges from 1.9% in Bihar to 86.84% in Chandigarh.
- Internet: Ranges from 1.9% in Bihar to 100% in states like Delhi and Lakshadweep.

Impact on Quality Education:

Electricity, computers, and internet access are fundamental for digital learning and integration of technology in education.

Suggestions for Improvement:

- Infrastructure Development: Invest in improving electricity supply and expand computer and internet access in underserved states.
- Digital Literacy: Provide training for students and teachers on using computers and internet resources effectively.

6. Drinking Water, Hand Wash Stations, and Ramps

Current Statistics:

- Drinking Water: Ranges from 24.66% in Meghalaya to 100% in states like Delhi and Lakshadweep.
- Hand Wash Stations: Ranges from 1.89% in Meghalaya to 100% in several states.
- Ramps: Ranges from 6.9% in Manipur to 100% in states like Delhi and Lakshadweep.

Impact on Quality Education:

Safe drinking water, hand wash stations, and ramps are crucial for health, hygiene, and accessibility. These facilities are essential for creating a conducive learning environment.

Suggestions for Improvement:

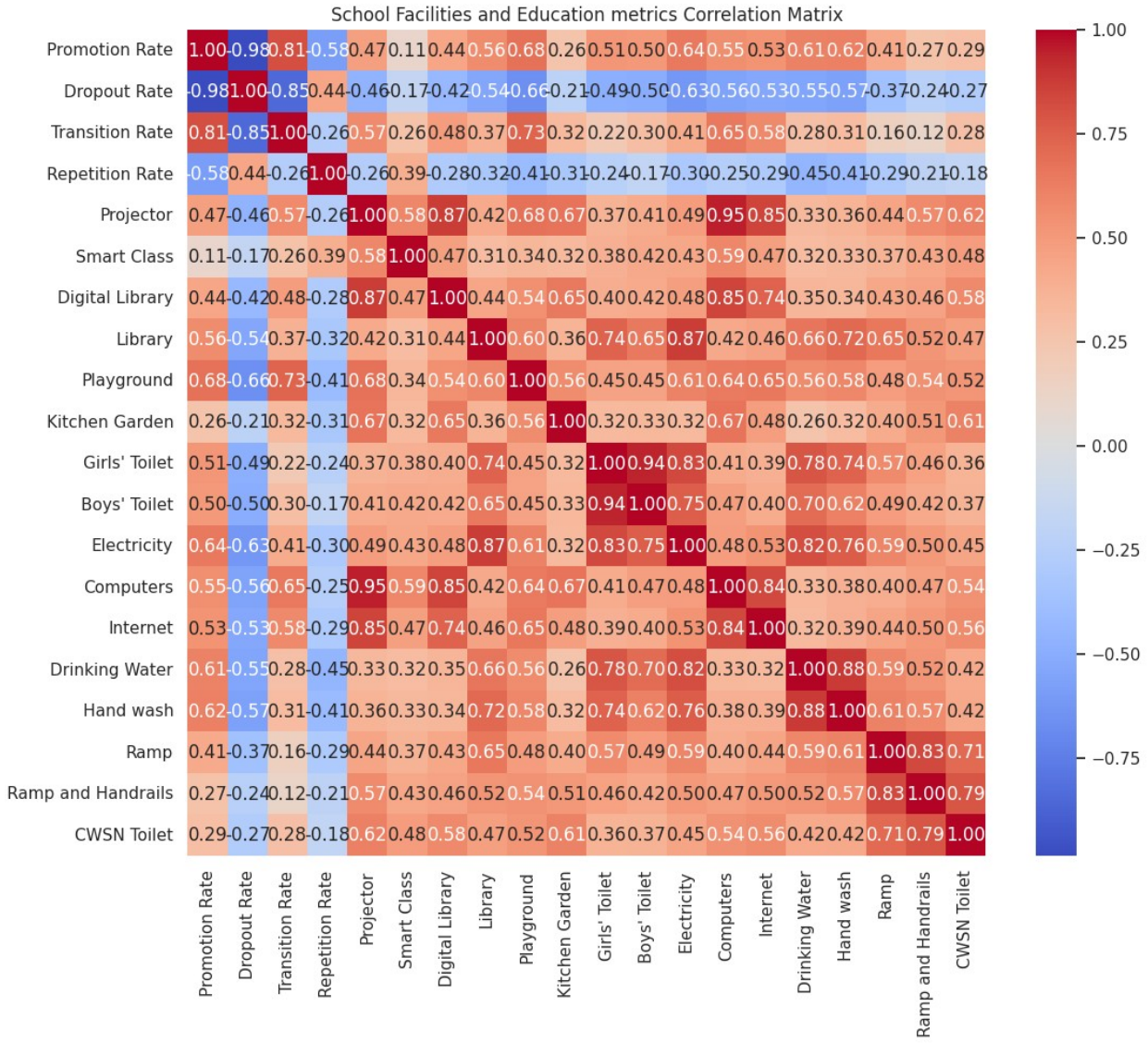
- Health and Accessibility: Ensure all schools are equipped with basic health and accessibility facilities, especially in states with lower percentages.
- Regular Maintenance: Implement regular checks and maintenance of these facilities to ensure they remain functional and effective.

Overall Recommendations

- Prioritize Infrastructure: Focus on states with low availability of critical facilities and invest in upgrading infrastructure.
- Implement Training Programs: Provide professional development for teachers and staff to maximize the impact of new technologies and facilities.
- Promote Inclusivity: Ensure all facilities support inclusive education, particularly for students with special needs.
- Monitor and Evaluate: Continuously monitor the availability and condition of facilities and adjust policies and investments as needed.
- By addressing these areas, educational outcomes can be significantly improved, leading to better student performance, engagement, and motivation across different states and UTs.

Analysing the correlation among facilities and education metrics

```
# Compute the correlation matrix
correlation_matrix =
final_set.drop(final_set.columns[[0,1,2]],axis=1).corr()

plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.2f')
plt.title('School Facilities and Education metrics Correlation
Matrix')
plt.show()
```



Analaysing the Correlation of facilities and the performance metrics

Facilities with Limited Impact on Student Performance and Quality Education

While a variety of facilities contribute to the quality of education, some have a less direct impact on overall student performance and quality compared to others. Based on general observations, the following facilities might not significantly influence key educational performance metrics (promotion rate, dropout rate, transition rate, and repetition rate):

- Kitchen Gardens: These can enhance environmental awareness and practical skills, but they are less likely to have a significant impact on core academic performance metrics.
- Ramps and ramps with handrails: Essential for accessibility, ramps are crucial for inclusion but may not directly influence academic metrics unless paired with other supportive educational infrastructure.
- CWSN Toilets: Necessary for accessibility, but their impact on academic performance is less direct. They are crucial for inclusivity but do not directly affect metrics.

Facilities with more direct impact on educational outcomes include:

- Projectors and Smart Classes: Enhance visual presentations and interactive lessons.mprove engagement and learning through interactive technology, significantly impacting educational outcomes.
- Digital Libraries: Provide access to a broad range of resources, supporting research and learning.
- Libraries: Offer essential resources for study and research, directly enhancing learning and academic performance.
- Playgrounds: Promote physical health and social skills, indirectly supporting overall educational development.
- Sanitation and Toilets: Essential for hygiene and inclusivity, indirectly supporting attendance and comfort.
- Electricity: Vital for the operation of educational technologies and tools, directly supporting effective learning environments.
- Computers: Facilitate digital learning and access to information, directly enhancing educational outcomes.
- Internet: Provides access to online resources and information, significantly impacting learning and educational engagement.
- Drinking Water: Ensures student health and hydration, indirectly supporting concentration and academic performance.

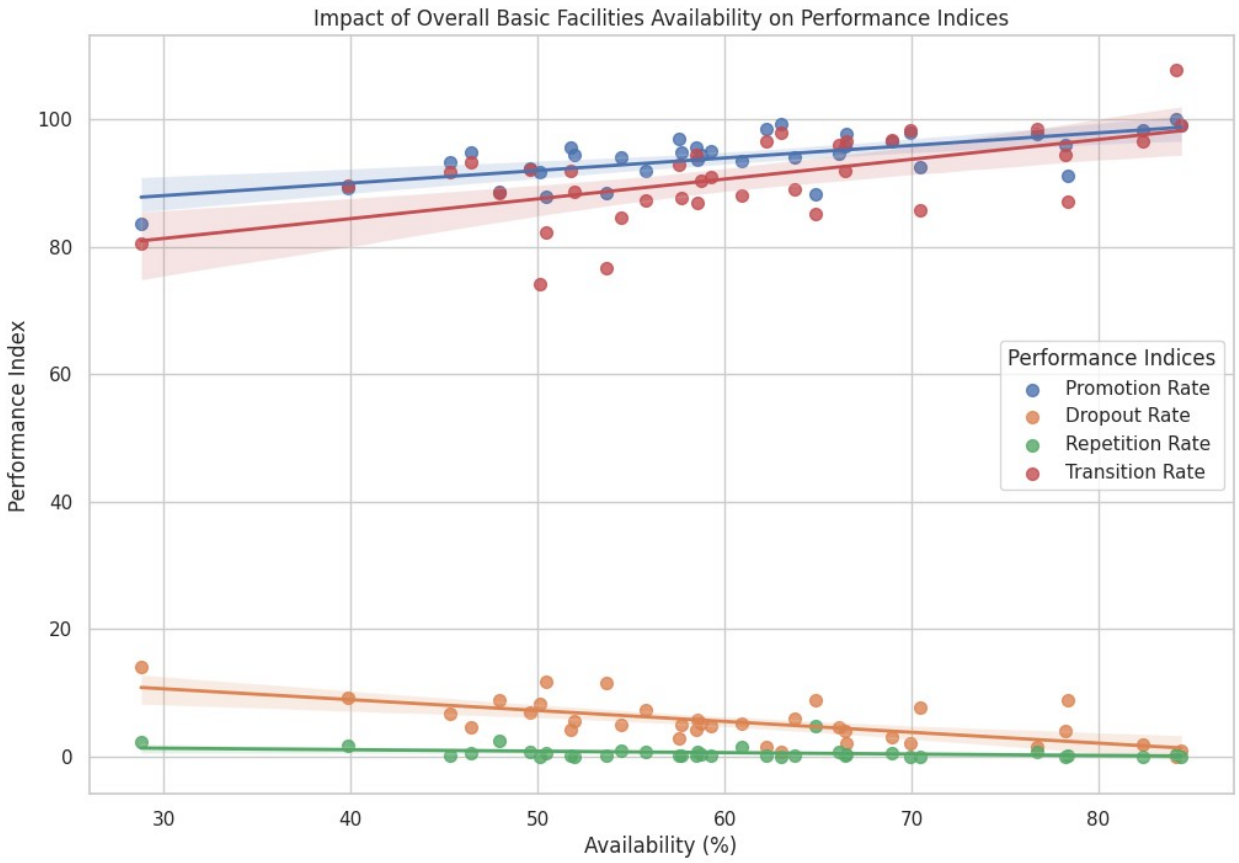
Overall, improved availability of educational facilities positively affects key performance metrics, contributing to better student outcomes and reduced educational setbacks.

Overall Analysis of Impact of Mean Basic Facilities

Lets prepare a new feature '**Mean Basic Facilities**' which wolud represent as the mean of availability of the infratech facilities having greater importance than others as mentioned before and study the trends in the education metrics with respect to it.

```
#Scatter Plot with Regression Line: Impact of Basic Facilities
Availability on Performance Indices
final_set['Mean Basic Facilities']=final_set[['Projector','Smart
Class','Digital Library','Library','Playground',"Girls' Toilet","Boys'
Toilet",'Electricity','Computers','Internet','Drinking Water','Hand
wash']].mean(axis=1)
performance_indices=['Promotion Rate','Dropout Rate','Repetition
Rate','Transition Rate']
plt.figure(figsize=(12, 8))
for index in performance_indices:
    sns.regplot(x='Mean Basic Facilities', y=index, data=final_set,
label=index, scatter_kws={'s':50}, line_kws={'linewidth':2})
plt.title('Impact of Overall Basic Facilities Availability on
Performance Indices')
plt.xlabel('Availability (%)')
plt.ylabel('Performance Index')
```

```
plt.legend(title='Performance Indices')
plt.show()
```



Analytical Summary of Linear Trend Models

1. Dropout Rate

- *Model Formula*: Dropout Rate = (Mean Basic Facilities + Intercept)
- *R-Squared*: 0.424
- *p-value*: < 0.0001 (significant)
- *Mean Basic Facilities*: -0.1705 (p < 0.0001)
- *Intercept*: 15.7729 (p < 0.0001)
- *Analysis*: The model explains 42.4% of the variance in dropout rates based on the availability of basic facilities. The negative coefficient for Mean Basic Facilities indicates that an increase in basic facilities is associated with a decrease in the dropout rate. This relationship is statistically significant, suggesting that improving basic facilities is likely to reduce dropout rates.

2. Promotion Rate

- *Model Formula*: Promotion Rate = (Mean Basic Facilities + Intercept)
- *R-Squared*: 0.438
- *p-value*: < 0.0001 (significant)

- *Mean Basic Facilities*: 0.1968 ($p < 0.0001$)
- *Intercept*: 82.0942 ($p < 0.0001$)
- *Analysis*: The model accounts for 43.8% of the variance in promotion rates based on basic facilities. The positive coefficient for Mean Basic Facilities suggests that more basic facilities are associated with higher promotion rates. This result is statistically significant, indicating a clear link between improved facilities and better promotion outcomes.

3. Repetition Rate

- *Model Formula*: Repetition Rate = (Mean Basic Facilities + Intercept)
- *R-Squared*: 0.0893
- *p-value*: 0.0812 (not significant)
- *Mean Basic Facilities*: -0.0227 ($p = 0.0812$)
- *Intercept*: 2.008 ($p = 0.0149$)
- *Analysis*: This model explains only 8.93% of the variance in repetition rates. The negative coefficient for Mean Basic Facilities suggests a potential decrease in repetition rates with increased facilities, but this relationship is not statistically significant at the 0.05 level. The impact of facilities on repetition rates may be less direct or influenced by other factors not captured in this model.

4. Transition Rate

- *Model Formula*: Transition Rate = (Mean Basic Facilities + Intercept)
- *R-Squared*: 0.345
- *p-value*: 0.0002 (significant)
- *Mean Basic Facilities*: 0.3090 ($p = 0.0002$)
- *Intercept*: 72.1433 ($p < 0.0001$)
- *Analysis*: The model explains 34.5% of the variance in transition rates. The positive coefficient for Mean Basic Facilities suggests that an increase in basic facilities is associated with higher transition rates. This result is statistically significant, indicating that improving facilities likely enhances the ability of students to transition smoothly between educational stages.

Overall Insights:

- *Significant Impact*: Basic facilities have a statistically significant impact on Dropout Rate, Transition Rate, and Promotion Rate. Improvements in basic facilities are generally associated with better educational outcomes in these areas.
- *Limited Impact*: The effect of basic facilities on Repetition Rate is not statistically significant, suggesting that other factors may play a more critical role in determining repetition rates.
- *R-Squared Values*: The models show moderate explanatory power for Dropout, Promotion, and Transition Rates, indicating that while basic facilities are important, they are part of a larger set of factors influencing educational performance.

Recommendations:

- *Focus on Facility Improvement*: Invest in upgrading basic facilities such as classrooms, libraries, and sanitation to potentially improve dropout and promotion rates.

- *Monitor and Evaluate*: Continuously monitor the impact of facility improvements and adjust strategies as needed to maximize educational outcomes.
- This summary provides a snapshot of how basic facilities correlate with educational performance metrics and highlights areas for potential focus and investment.

Prediction Model

In our model, we are going to use the facilities with more direct impact on the educational outcomes for predicting the important or significant educational metrics. Lets choose the **Promotion Rate** and **Dropout Rate** as the important metrics to be predicted in our model.

Split the Data frame into Features and Targets

```
# FEATURES
data = final_set
features = data[['Projector', 'Smart Class', 'Digital
Library','Library',
                 'Computers', 'Internet', 'Playground',
                 "Girls' Toilet", "Boys' Toilet",
                 'Electricity', 'Drinking Water',
                 'Hand wash']]

# TARGETS
promotion_rate = data['Promotion Rate']
dropout_rate = data['Dropout Rate']
```

Train and test the data

```
# Train and Test Data
X_train, X_test, y_train_prom, y_test_prom =
train_test_split(features, promotion_rate, test_size=0.2,
random_state=42)
_, _, y_train_drop, y_test_drop = train_test_split(features,
dropout_rate, test_size=0.2, random_state=42)
```

Selection of Appropriate Algorithm

Given that our project aims to predict promotion and dropout rates based on the availability percentage of different school facilities, the choice of algorithm should consider both the nature of the data and the problem. Since we are dealing with regression tasks, lets check the most suitable algorithms for our model, along with their characteristics:

```
# Lets consider only Promotion Rate for model evaluation purpose
# Initialize the models
models = {
    "Linear Regression": LinearRegression(),
    "Decision Tree": DecisionTreeRegressor(),
```

```
    "Support Vector Regressor": SVR(),
    "Random Forest": RandomForestRegressor(),
    "Gradient Boosting": GradientBoostingRegressor(),
    "K-Nearest Neighbors": KNeighborsRegressor(),
    "Elastic Net": ElasticNet()
}

# Evaluate each model
for name, model in models.items():
    model.fit(X_train, y_train_prom)
    y_pred = model.predict(X_test)

    mae = mean_absolute_error(y_test_prom, y_pred)
    mse = mean_squared_error(y_test_prom, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test_prom, y_pred)

    print(f'{name} Evaluation:')
    print(f'MAE: {mae}')
    print(f'MSE: {mse}')
    print(f'RMSE: {rmse}')
    print(f'R2: {r2}\n')
```

```
Linear Regression Evaluation:
MAE: 3.041158655730614
MSE: 11.264689889562717
RMSE: 3.3562910913034223
R2: 0.010657942475235105

Decision Tree Evaluation:
MAE: 3.5083333333333346
MSE: 19.16472222222221
RMSE: 4.377753102017313
R2: -0.6831768917839183

Support Vector Regressor Evaluation:
MAE: 2.6061766363732506
MSE: 10.464350697929119
RMSE: 3.2348648654818826
R2: 0.08094920040876619

Random Forest Evaluation:
MAE: 2.808583333333347
MSE: 10.454387611111224
RMSE: 3.233324544661613
R2: 0.08182422678839563

Gradient Boosting Evaluation:
MAE: 3.2223770402148357
MSE: 14.152748800734514
```

```
RMSE: 3.762013928833134
R2: -0.24299113028608832

K-Nearest Neighbors Evaluation:
MAE: 3.1700000000000053
MSE: 13.935377777777813
RMSE: 3.7330118909237098
R2: -0.22390012137180793

Elastic Net Evaluation:
MAE: 2.3882230409221066
MSE: 7.160106510550545
RMSE: 2.6758375344087213
R2: 0.37115050865199406
```

Model Evaluation Analysis

Based on the model evaluation results, let's analyze and select the most appropriate model for predicting the promotion rate based on the given metrics.

Metrics Overview:

- MAE (Mean Absolute Error): Lower values indicate better performance.
- MSE (Mean Squared Error): Lower values indicate better performance.
- RMSE (Root Mean Squared Error): Lower values indicate better performance.
- R2 (R-squared): Values closer to 1 indicate better performance.

Recommendation:

- *Elastic Net:* This model shows the best overall performance with the lowest MAE, MSE, RMSE, and the highest R-squared value. It indicates the most accurate predictions and the best fit to the data.
- *Random Forest:* This model is great candidate, with competitive performance across all metrics. It handles non-linear relationships well and provides feature importance.

Elastic Net Model and Evaluation

```
# Train Elastic Net Model for Promotion Rate
elastic_net_prom = ElasticNet(random_state=42)
elastic_net_prom.fit(X_train, y_train_prom)

# Predictions
y_pred_prom = elastic_net_prom.predict(X_test)

# Evaluation for Promotion Rate
mae_prom = mean_absolute_error(y_test_prom, y_pred_prom)
mse_prom = mean_squared_error(y_test_prom, y_pred_prom)
rmse_prom = np.sqrt(mse_prom)
r2_prom = r2_score(y_test_prom, y_pred_prom)
```

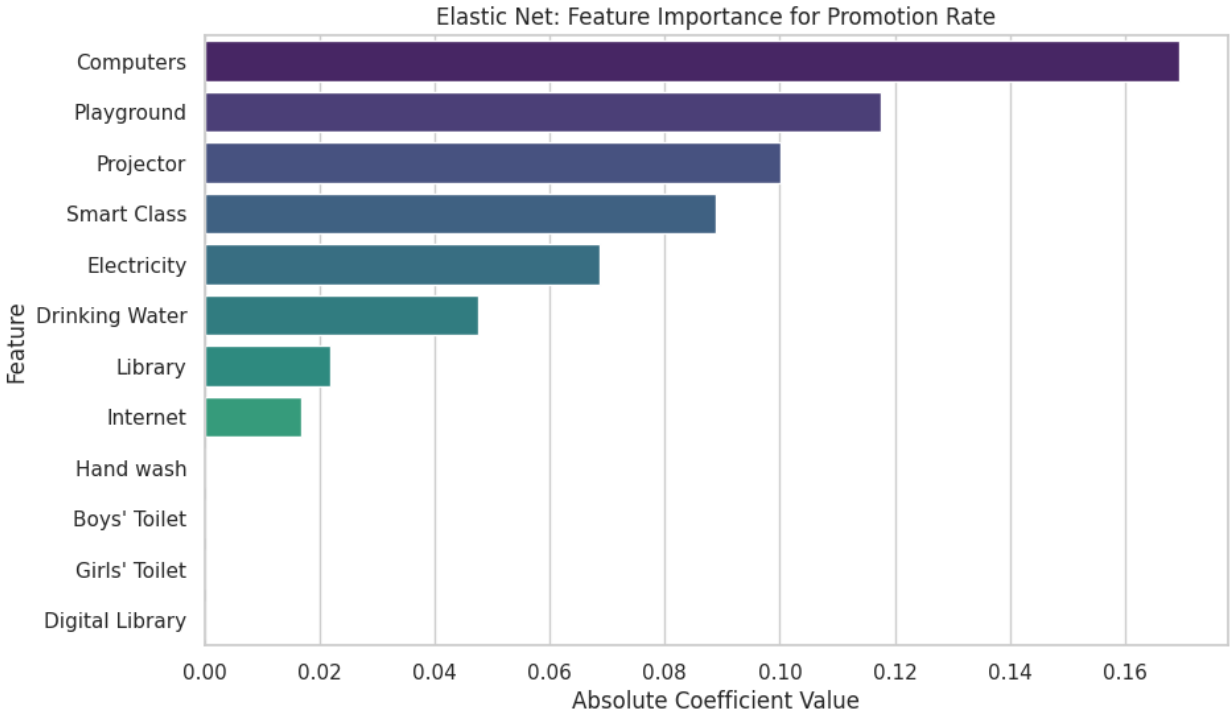
```
print("Elastic Net Promotion Rate Evaluation:")
print(f"MAE: {mae_prom}")
print(f"MSE: {mse_prom}")
print(f"RMSE: {rmse_prom}")
print(f"R2: {r2_prom}")

# Train Elastic Net Model for Dropout Rate
elastic_net_drop = ElasticNet(random_state=42)
elastic_net_drop.fit(X_train, y_train_drop)

# Predictions
y_pred_drop = elastic_net_drop.predict(X_test)

# Evaluation for Dropout Rate
mae_drop = mean_absolute_error(y_test_drop, y_pred_drop)
mse_drop = mean_squared_error(y_test_drop, y_pred_drop)
rmse_drop = np.sqrt(mse_drop)
r2_drop = r2_score(y_test_drop, y_pred_drop)

print("\nElastic Net Dropout Rate Evaluation:")
print(f"MAE: {mae_drop}")
print(f"MSE: {mse_drop}")
print(f"RMSE: {rmse_drop}")
print(f"R2: {r2_drop}")
```

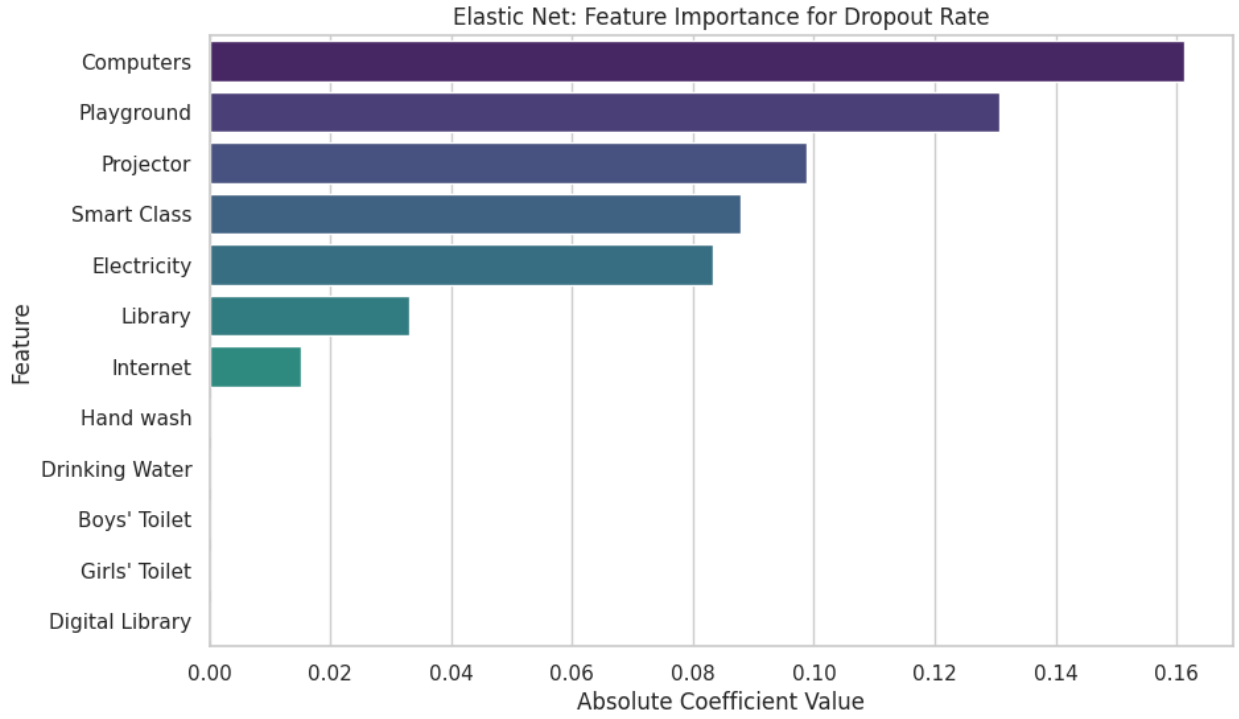
```
Elastic Net Promotion Rate Evaluation:
MAE: 2.3882230409221066
MSE: 7.160106510550545
RMSE: 2.6758375344087213
R2: 0.37115050865199406

Elastic Net Dropout Rate Evaluation:
MAE: 2.5653386557510585
MSE: 9.118046802105978
RMSE: 3.019610372565636
R2: -0.31864584031440724
```

Visual Analysis of Elastic Net Model

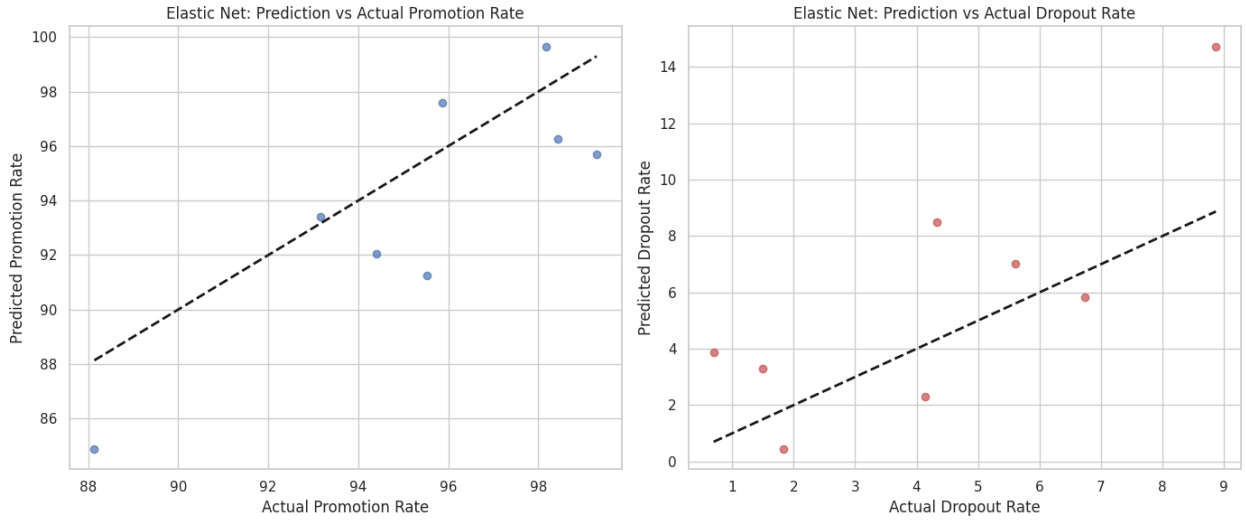
```
# Feature Importance for Promotion Rate
plt.figure(figsize=(10, 6))
importance_prom = np.abs(elastic_net_prom.coef_)  # Absolute value of
coefficients
indices_prom = np.argsort(importance_prom)[::-1]
sns.barplot(x=importance_prom[indices_prom],
y=features.columns[indices_prom], palette='viridis')
plt.xlabel('Absolute Coefficient Value')
plt.ylabel('Feature')
plt.title('Elastic Net: Feature Importance for Promotion Rate')
```

```
plt.show()

# Feature Importance for Dropout Rate
plt.figure(figsize=(10, 6))
importance_drop = np.abs(elastic_net_drop.coef_)  # Absolute value of
coefficients
indices_drop = np.argsort(importance_drop)[::-1]
sns.barplot(x=importance_drop[indices_drop],
y=features.columns[indices_drop], palette='viridis')
plt.xlabel('Absolute Coefficient Value')
plt.ylabel('Feature')
plt.title('Elastic Net: Feature Importance for Dropout Rate')
plt.show()
```

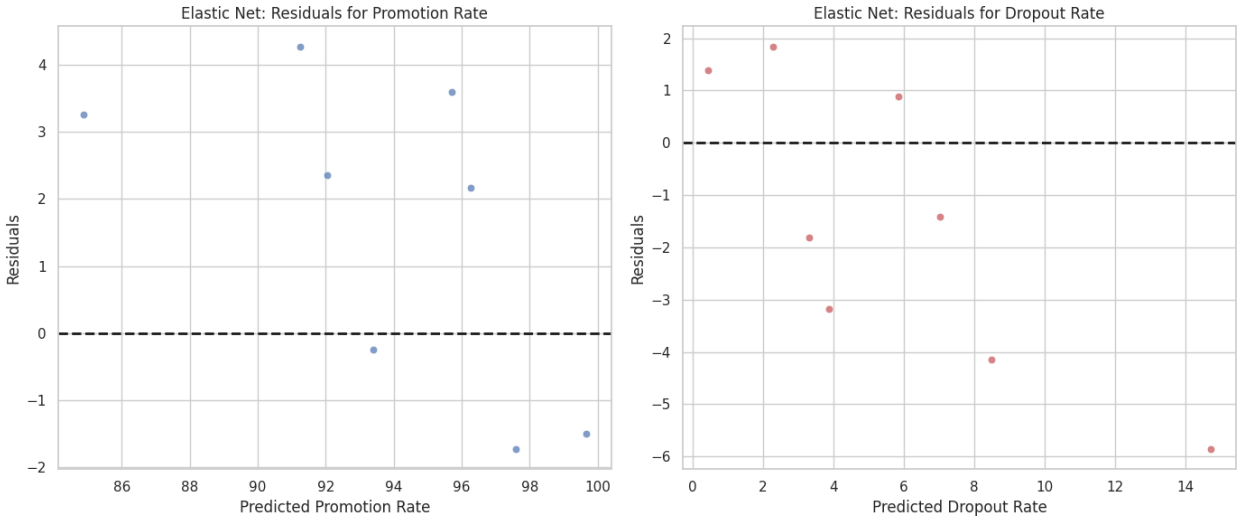





```
# Prediction vs Actual for Promotion Rate
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.scatter(y_test_prom, y_pred_prom, alpha=0.7, color='b')
plt.plot([y_test_prom.min(), y_test_prom.max()], [y_test_prom.min(),
y_test_prom.max()], 'k--', lw=2)
plt.xlabel('Actual Promotion Rate')
plt.ylabel('Predicted Promotion Rate')
plt.title('Elastic Net: Prediction vs Actual Promotion Rate')

# Prediction vs Actual for Dropout Rate
y_pred_drop = elastic_net_drop.predict(X_test)

plt.subplot(1, 2, 2)
plt.scatter(y_test_drop, y_pred_drop, alpha=0.7, color='r')
plt.plot([y_test_drop.min(), y_test_drop.max()], [y_test_drop.min(),
y_test_drop.max()], 'k--', lw=2)
plt.xlabel('Actual Dropout Rate')
plt.ylabel('Predicted Dropout Rate')
plt.title('Elastic Net: Prediction vs Actual Dropout Rate')

plt.tight_layout()
plt.show()
```




```
# Residuals for Promotion Rate
residuals_prom = y_test_prom - y_pred_prom

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=y_pred_prom, y=residuals_prom, color='b', alpha=0.7)
plt.axhline(0, color='k', linestyle='--', lw=2)
plt.xlabel('Predicted Promotion Rate')
plt.ylabel('Residuals')
plt.title('Elastic Net: Residuals for Promotion Rate')

# Residuals for Dropout Rate
residuals_drop = y_test_drop - y_pred_drop

plt.subplot(1, 2, 2)
sns.scatterplot(x=y_pred_drop, y=residuals_drop, color='r', alpha=0.7)
plt.axhline(0, color='k', linestyle='--', lw=2)
plt.xlabel('Predicted Dropout Rate')
plt.ylabel('Residuals')
plt.title('Elastic Net: Residuals for Dropout Rate')

plt.tight_layout()
plt.show()
```



Random Forest Model and Evaluation

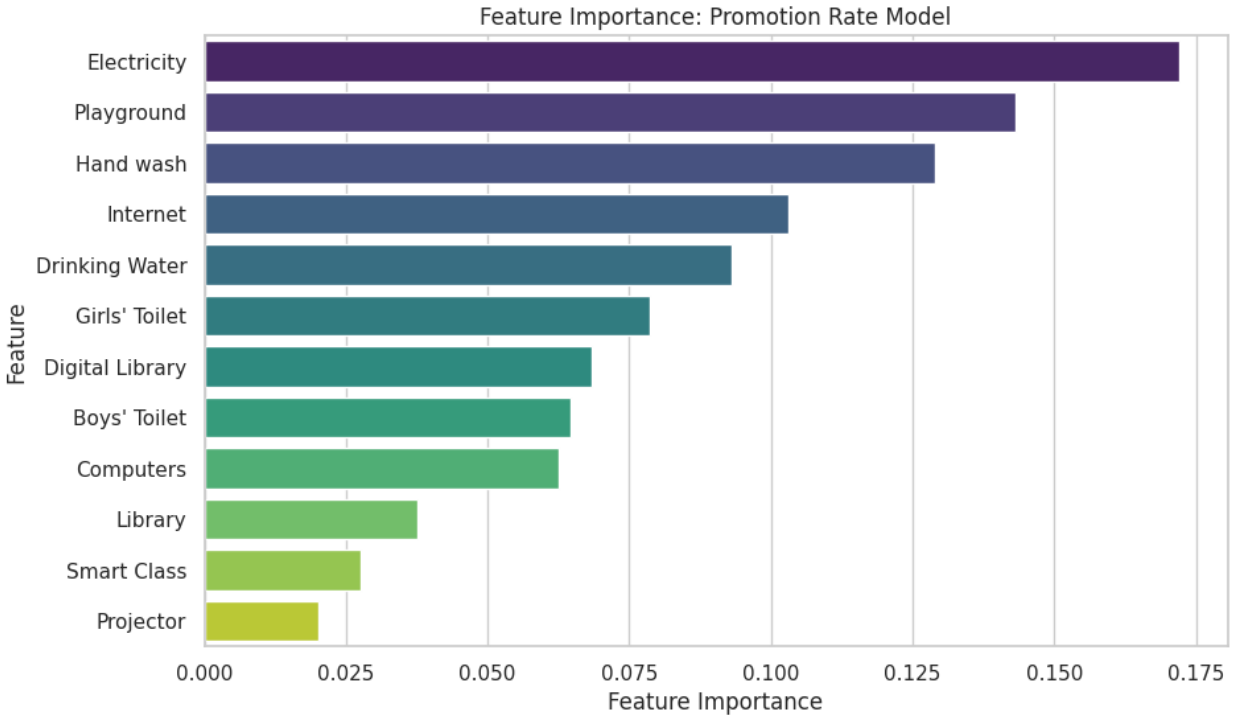
```
# Train the models
prom_model = RandomForestRegressor()
drop_model = RandomForestRegressor()

prom_model.fit(X_train, y_train_prom)
drop_model.fit(X_train, y_train_drop)

# Predictions
y_pred_prom = prom_model.predict(X_test)
y_pred_drop = drop_model.predict(X_test)

# Evaluation metrics for promotion rate model
mae_prom = mean_absolute_error(y_test_prom, y_pred_prom)
mse_prom = mean_squared_error(y_test_prom, y_pred_prom)
rmse_prom = np.sqrt(mse_prom)
r2_prom = r2_score(y_test_prom, y_pred_prom)

print(f'Promotion Rate Model Evaluation:')
print(f'MAE: {mae_prom}')
print(f'MSE: {mse_prom}')
print(f'RMSE: {rmse_prom}')
print(f'R2: {r2_prom}')

# Evaluation metrics for dropout rate model
mae_drop = mean_absolute_error(y_test_drop, y_pred_drop)
mse_drop = mean_squared_error(y_test_drop, y_pred_drop)
rmse_drop = np.sqrt(mse_drop)
r2_drop = r2_score(y_test_drop, y_pred_drop)

print(f'\nDropout Rate Model Evaluation:')
print(f'MAE: {mae_drop}')
print(f'MSE: {mse_drop}')
```

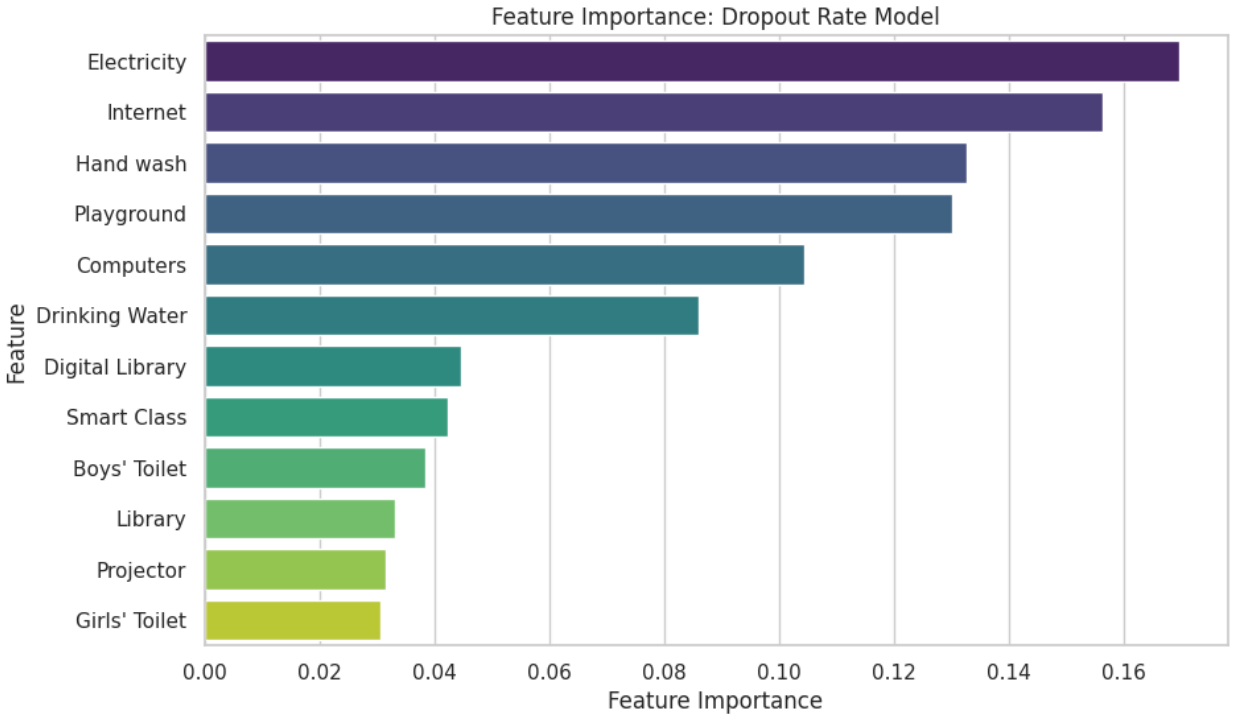
```
print(f'RMSE: {rmse_drop}')
print(f'R2: {r2_drop}')
```

```
Promotion Rate Model Evaluation:
MAE: 2.9502083333333466
MSE: 11.004039152777844
RMSE: 3.3172336596594825
R2: 0.033550071664252945

Dropout Rate Model Evaluation:
MAE: 1.976750000000001
MSE: 6.206771722222224
RMSE: 2.491339343048679
R2: 0.1023808178524529
```

Visual Analysis of Random Forest Regressor Model

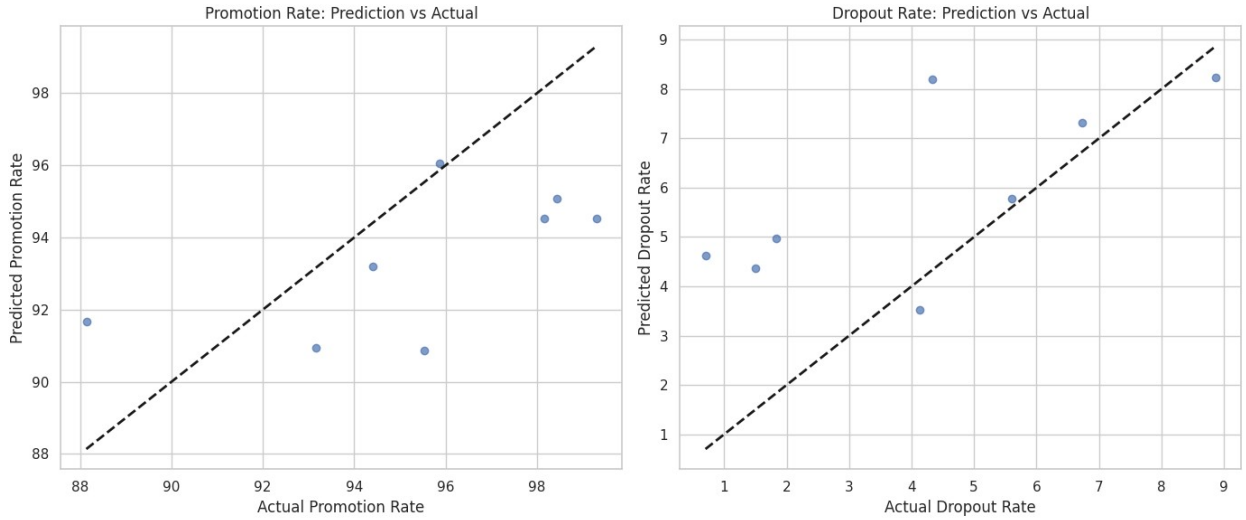
```
# Feature Importance for Promotion Rate Model
importances_prom = prom_model.feature_importances_
indices_prom = np.argsort(importances_prom)[::-1]

plt.figure(figsize=(10, 6))
sns.barplot(x=importances_prom[indices_prom],
y=features.columns[indices_prom], palette='viridis')
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.title('Feature Importance: Promotion Rate Model')
plt.show()

# Feature Importance for Dropout Rate Model
importances_drop = drop_model.feature_importances_
indices_drop = np.argsort(importances_drop)[::-1]

plt.figure(figsize=(10, 6))
sns.barplot(x=importances_drop[indices_drop],
y=features.columns[indices_drop], palette='viridis')
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.title('Feature Importance: Dropout Rate Model')
plt.show()
```





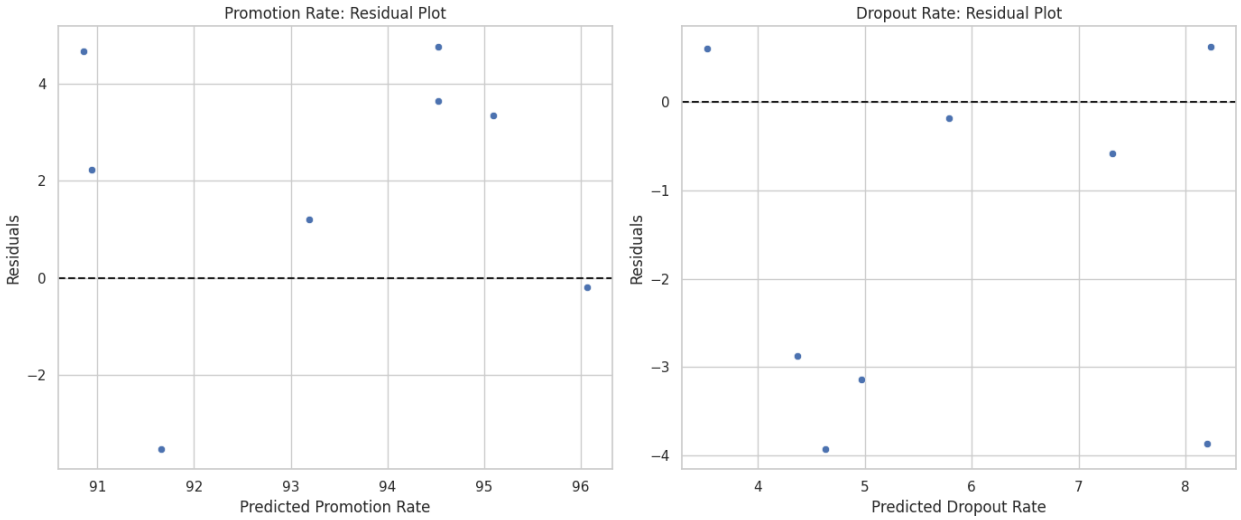

```
# Prediction vs Actual for Promotion Rate Model
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.scatter(y_test_prom, y_pred_prom, alpha=0.7)
```

```
plt.plot([y_test_prom.min(), y_test_prom.max()], [y_test_prom.min(),
y_test_prom.max()], 'k--', lw=2)
plt.xlabel('Actual Promotion Rate')
plt.ylabel('Predicted Promotion Rate')
plt.title('Promotion Rate: Prediction vs Actual')

# Prediction vs Actual for Dropout Rate Model
plt.subplot(1, 2, 2)
plt.scatter(y_test_drop, y_pred_drop, alpha=0.7)
plt.plot([y_test_drop.min(), y_test_drop.max()], [y_test_drop.min(),
y_test_drop.max()], 'k--', lw=2)
plt.xlabel('Actual Dropout Rate')
plt.ylabel('Predicted Dropout Rate')
plt.title('Dropout Rate: Prediction vs Actual')

plt.tight_layout()
plt.show()
```




```
# Residuals for Promotion Rate Model
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
residuals_prom = y_test_prom - y_pred_prom
sns.scatterplot(x=y_pred_prom, y=residuals_prom)
plt.axhline(0, color='k', linestyle='--')
plt.xlabel('Predicted Promotion Rate')
plt.ylabel('Residuals')
plt.title('Promotion Rate: Residual Plot')

# Residuals for Dropout Rate Model
plt.subplot(1, 2, 2)
residuals_drop = y_test_drop - y_pred_drop
sns.scatterplot(x=y_pred_drop, y=residuals_drop)
```

```
plt.axhline(0, color='k', linestyle='--')
plt.xlabel('Predicted Dropout Rate')
plt.ylabel('Residuals')
plt.title('Dropout Rate: Residual Plot')

plt.tight_layout()
plt.show()
```



Explanation for Visual Analysis of Model Characteristics:

- *Prediction vs. Actual Plot:* Scatter plots compare the actual values with the predicted values. The closer the points are to the diagonal line, the better the model's performance.
- *Residual Plot:* Scatter plots of the residuals (errors) vs. the predicted values help to check if there are any patterns in the errors. Ideally, the points should be randomly scattered around the horizontal line (residual = 0).
- *Feature Importance Plot:* Bar plots of the feature coefficients indicate the importance of each feature. Features with larger absolute coefficient values are considered more important.

Both the models align with the desired characteristics.

Significance of the Prediction Model for the Project

In the context of our project aimed at improving educational infrastructure and technology to enhance student performance and align with Sustainable Development Goal 4 (SDG 4), the predictive models we've developed play a crucial role. Here's how they contribute to your project's objectives:

1. Understanding the Impact of Infrastructure on Student Performance

Objective: To develop predictive models for student performance based on infrastructure quality.

Utility:

- Quantifying Relationships: The prediction models (Random Forest and Elastic Net) analyze how various elements of school infrastructure and digital initiatives influence key outcomes like promotion rates and dropout rates.
- Identifying Key Factors: By understanding which infrastructure elements (e.g., availability of smart classrooms, digital libraries, sanitation facilities) most significantly impact student outcomes, the models help in identifying critical areas for intervention.

2. Targeted Interventions for Improvement

Objective: To propose actionable solutions and policy recommendations to improve educational facilities.

Utility:

- Data-Driven Recommendations: The insights from these models provide a data-driven basis for recommending specific improvements. For instance, if the models reveal that the lack of a digital library is strongly associated with lower promotion rates, targeted investments in digital libraries can be prioritized.
- Resource Allocation: The models can guide how to allocate resources more effectively by highlighting which infrastructure improvements will yield the greatest benefit for student performance.

3. Assessing Regional Disparities

Objective: To understand regional disparities in educational infrastructure and technology.

Utility:

- Regional Analysis: By applying the models to different regions, we can assess how regional variations in infrastructure affect student outcomes. This can reveal disparities and highlight regions that need urgent attention.
- Customized Solutions: Insights from regional analyses allow for the development of customized solutions tailored to the specific needs and challenges of each region.

4. Evaluating the Effectiveness of Proposed Solutions

Objective: To assess the potential impact of these solutions on achieving SDG 4.

Utility:

- Impact Assessment: The predictive models can simulate the potential impact of proposed infrastructure improvements on student outcomes. This helps in evaluating whether the solutions will likely contribute to the achievement of SDG 4 by improving educational quality.
- Scenario Analysis: By running different scenarios (e.g., what if we increase digital library availability by 20%?), you can forecast the possible effects of various interventions and prioritize actions based on projected outcomes.

Summary

The prediction model is a key tool in our project's strategy to enhance educational infrastructure and align with SDG 4. They offer valuable insights into the relationship between infrastructure quality and student performance, guide targeted interventions, assess regional disparities, and evaluate the potential impact of proposed improvements. By leveraging these models, our project can ensure that investments in educational facilities are both effective and aligned with the goal of providing inclusive and equitable quality education.

Conclusion

The predictive models offer significant value by highlighting the relative impact of educational infrastructure on student performance. They help identify key areas for improvement, guide targeted interventions, and address regional disparities. However, these models are just one piece of the puzzle. A comprehensive approach to enhancing education must also consider other critical factors that influence student outcomes. By integrating these insights into a broader strategy, we can better support the goal of providing inclusive and equitable quality education as outlined in SDG 4.