

✖ Import Required Libraries


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import KMeans
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error




sns.set(style='whitegrid')
```

✖ Load Sample Data

```
np.random.seed(42)
regions = [f"Region_{i}" for i in range(1, 11)]
data = {
    "region": regions,
    "infra_access_score": np.random.randint(20, 81, size=10),
    "internet_penetration": np.round(np.random.uniform(0.2, 0.9, size=10), 2),
    "startup_density": np.random.randint(0, 31, size=10),
    "education_index": np.round(np.random.uniform(0.4, 0.9, size=10), 2),
    "unemployment_rate": np.round(np.random.uniform(0.05, 0.25, size=10), 2)
}
df = pd.DataFrame(data)
df
```



	region	infra_access_score	internet_penetration	startup_density	education_index	unemployment_rate
0	Region_1	58	0.31	20	0.71	0.17
1	Region_2	71	0.24	0	0.47	0.08
2	Region_3	48	0.81	11	0.55	0.06
3	Region_4	34	0.62	25	0.58	0.24
4	Region_5	62	0.70	21	0.63	0.24
5	Region_6	27	0.21	28	0.79	0.21
6	Region_7	80	0.88	11	0.50	0.11
7	Region_8	40	0.78	24	0.66	0.07
8	Region_9	58	0.35	16	0.70	0.19
9	Region_10	77	0.33	26	0.42	0.14

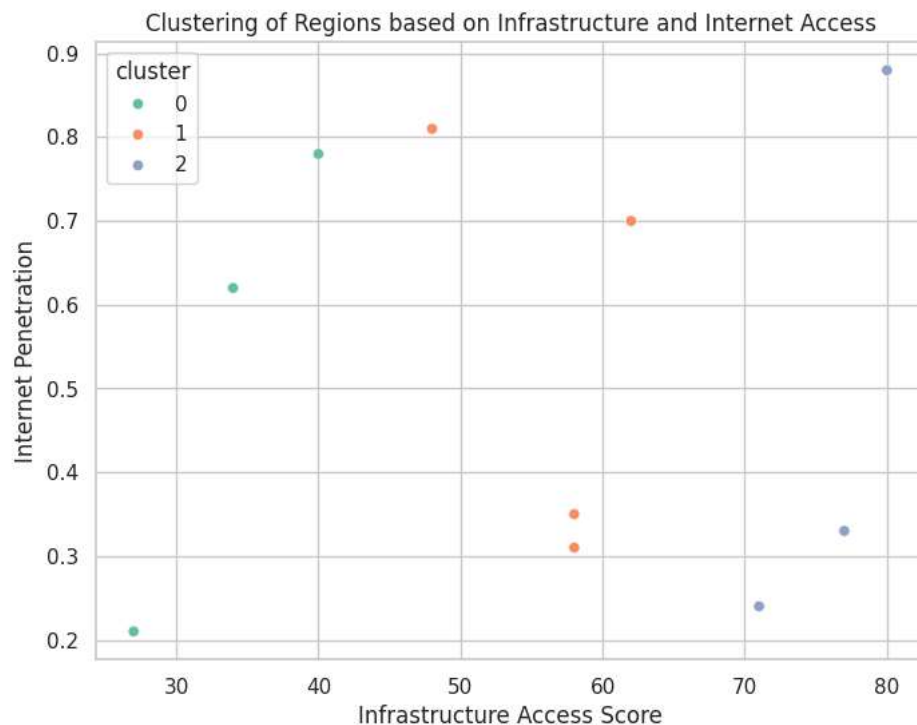


Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

✖ Cluster Underserved Regions

```
X_cluster = df[['infra_access_score', 'internet_penetration']]
kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(X_cluster)

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='infra_access_score', y='internet_penetration', hue='cluster', palette='Set2')
plt.title("Clustering of Regions based on Infrastructure and Internet Access")
plt.xlabel("Infrastructure Access Score")
plt.ylabel("Internet Penetration")
plt.show()
```



✓ Calculate Innovation Index

```
scaler = MinMaxScaler()
df[['startup_density_norm', 'education_index_norm']] = scaler.fit_transform(df[['startup_density', 'education_index']])
df['innovation_index'] = (df['startup_density_norm'] + df['education_index_norm']) / 2
df[['region', 'innovation_index']]
```



	region	innovation_index
0	Region_1	0.749035
1	Region_2	0.067568
2	Region_3	0.372104
3	Region_4	0.662645
4	Region_5	0.658784
5	Region_6	1.000000
6	Region_7	0.304537
7	Region_8	0.752896
8	Region_9	0.664093
9	Region_10	0.464286


✓ Predict Economic Upliftment




```
df['economic_upliftment'] = (0.6 * df['infra_access_score'] + 0.4 * df['innovation_index'] * 100) + np.random.normal(0, 5, size=10)
```

```
X = df[['infra_access_score', 'innovation_index']]
y = df['economic_upliftment']
```

```
model = GradientBoostingRegressor(random_state=42)
model.fit(X, y)
df['upliftment_predicted'] = model.predict(X)
```

```
df_sorted = df[['region', 'upliftment_predicted']].sort_values(by='upliftment_predicted', ascending=False)
df_sorted.reset_index(drop=True, inplace=True)
df_sorted
```



	region	upliftment_predicted	
0	Region_10	65.626541	
1	Region_9	65.052917	
2	Region_1	64.695617	
3	Region_5	64.597640	
4	Region_8	55.098263	
5	Region_7	53.541333	
6	Region_3	47.796640	
7	Region_6	46.402616	
8	Region_4	40.802199	
9	Region_2	40.015055	

Next steps: [Generate code with df_sorted](#) [View recommended plots](#) [New interactive sheet](#)