Sudanese Primary Schools Dataset Analysis and Classification based on Facility Availability

Data Preparation/Feature Engineering

1. Overview

The main aim of this project is to:

- analyze Sudan school's dataset and visualize it
- clustering
- classification of schools according to facilities

The project contains many parts.

1. Dataset
    - Dataset preparation
    - Exploratory data analysis (EDA)
    - Prepare dataset for machine learning model (Feature engineering)
2. Clustering
    - K-means clustering
3. Classification
    - Machine learning models (random forest, knn, svm, naive bayes, nn)
    - Comparison

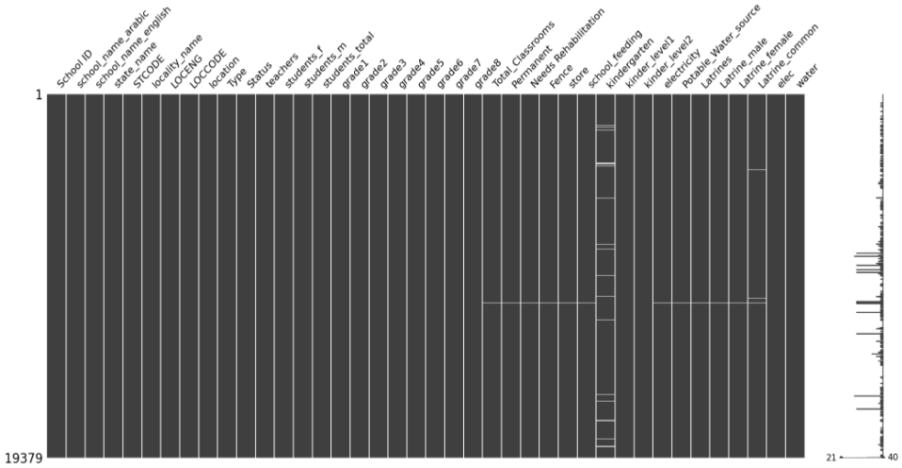
2. Data Collection

The project aims to analyze Sudanese schools' dataset and classify it according to the facilities to enhance the understanding of the current state of Sudanese schools. As known, Sudan one of the least development countries which still need many steps of development in many aspects but the most important is to provide well based education environment and suffers from proper dataset that can help in analyzing the situation and making good decisions. An impressive collaboration between the Ministry of Education, UNICEF, and OCHA Sudan have been done in 2021 in collecting the school's data in order to ease the decision-making process and to know the actual statics of students, teachers and facilities availability as well.

3. Data Cleaning

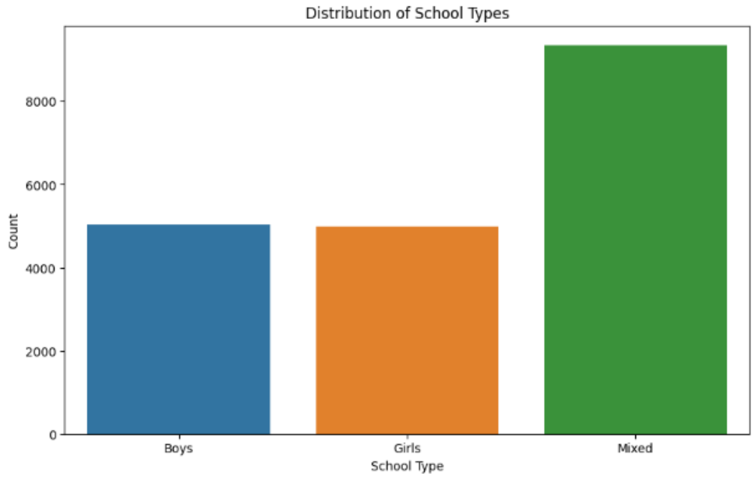
As the dataset is quite big, dropping all missing data rows seems a suitable solution. Dataset size changed from 19379 ==> 18716 only 663 rows have been dropped. Below is visualization of missing data distribution across dataset.

```
School ID                  0
school_name_arabic        16
school_name_english       12
state_name                13
STCODE                    12
locality_name             13
LOCENG                    16
LOCCODE                   16
location                   2
Type                       4
Status                     4
teachers                   0
students_f                 0
students_m                 0
students_total             0
grade1                     0
grade2                     0
grade3                     0
grade4                     0
grade5                     0
grade6                     0
grade7                     0
grade8                     0
Total_Classrooms          52
Permanent                 53
Needs Rehabilitation      52
Fence                     46
store                     44
school_feeding            44
kindergarten             506
kinder_level1              5
kinder_level2              7
electricity               44
Potable_Water_source      44
Latrines                  45
Latrine_male              67
Latrine_female            75
Latrine_common            85
```

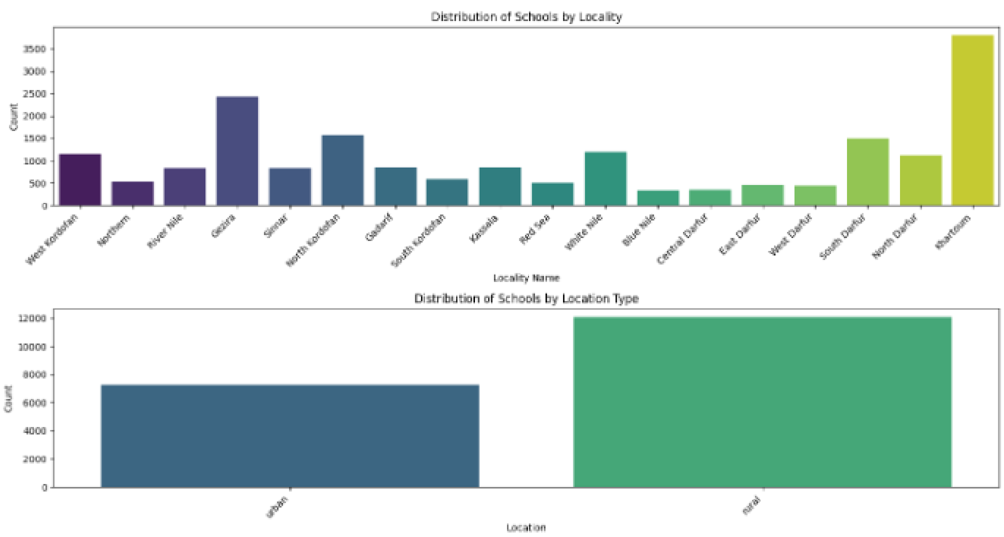


4. Exploratory Data Analysis (EDA)

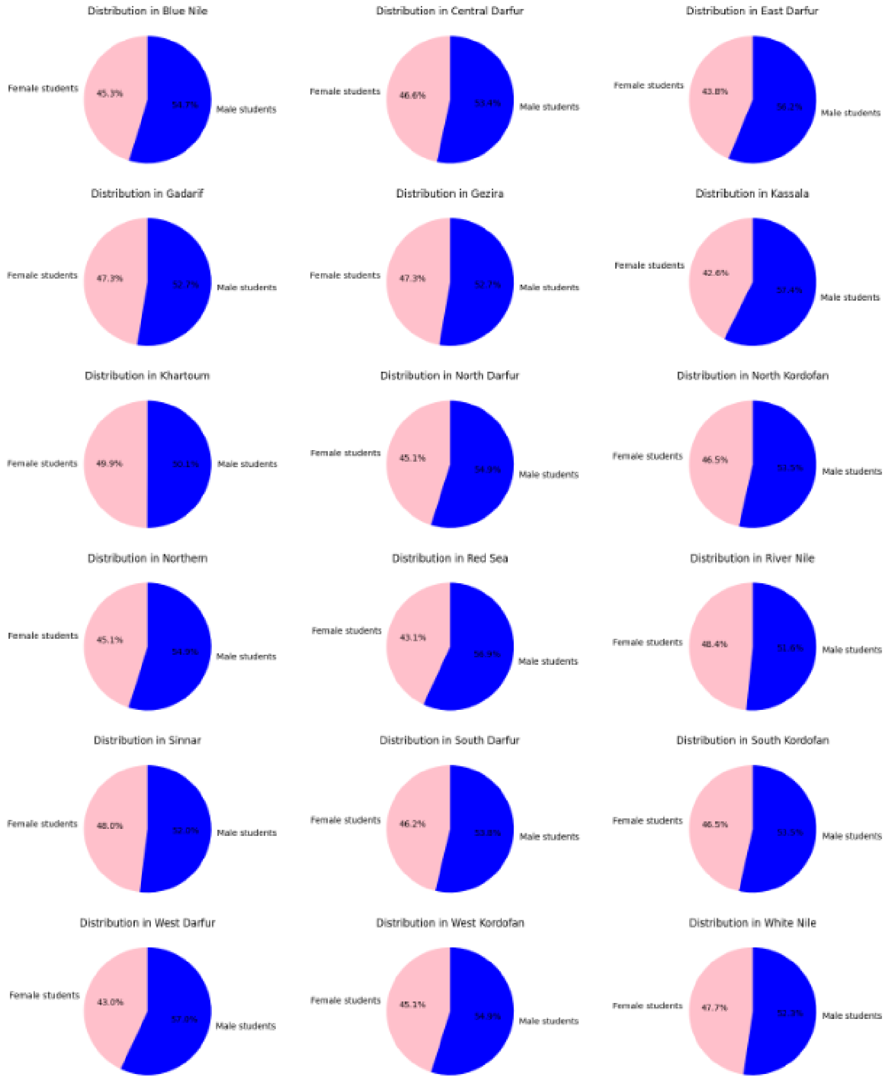
4.1. Distribution of Sudanese school types



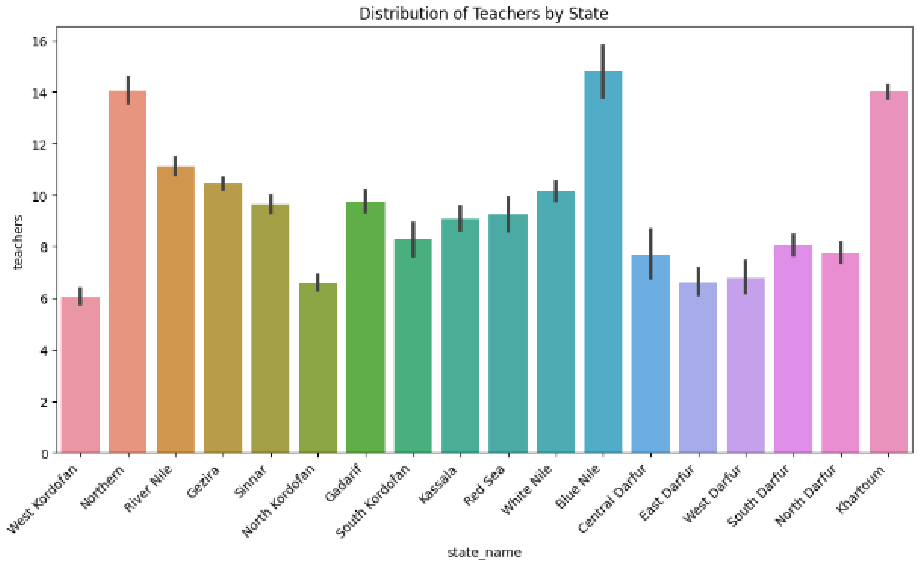
4.2. Distribution of Sudanese by locality and areal type (urban, rural)



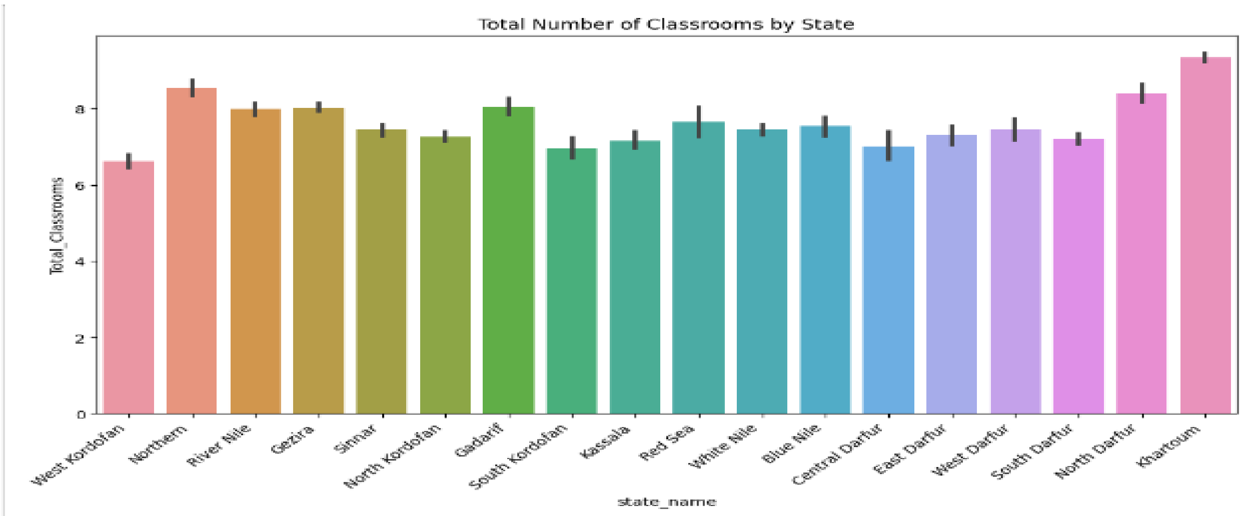
4.3. Distribution of Sudanese school male-female across states



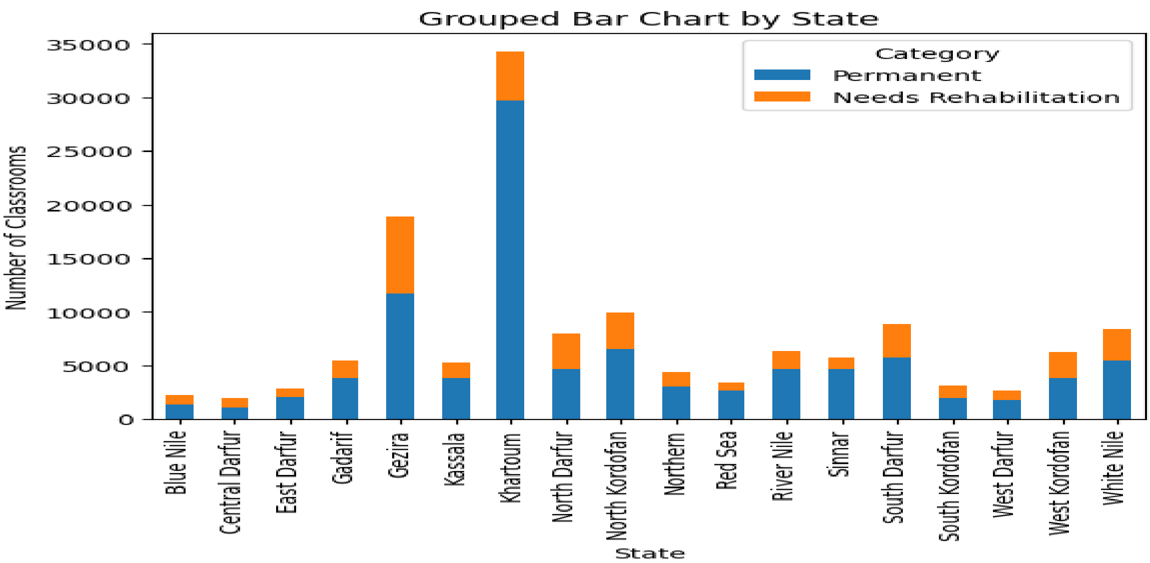
4.4. Distribution of teachers across different states



4.5. *Total number of classrooms in each state*



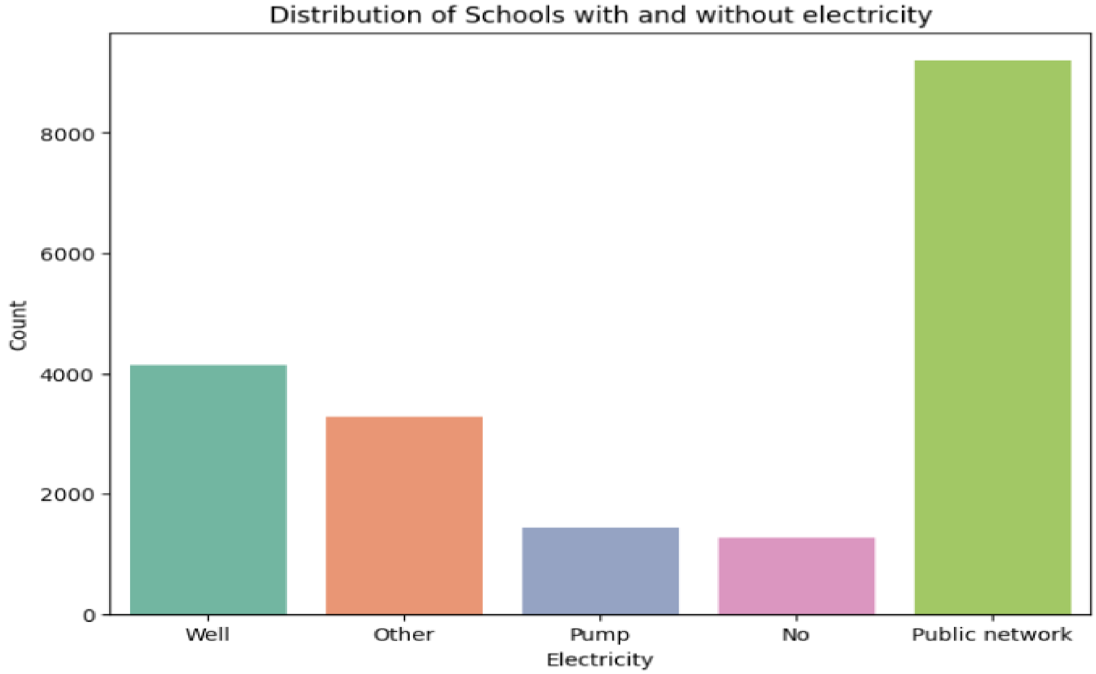
4.6. *Classes status based on permanent or need rehabilitation.*

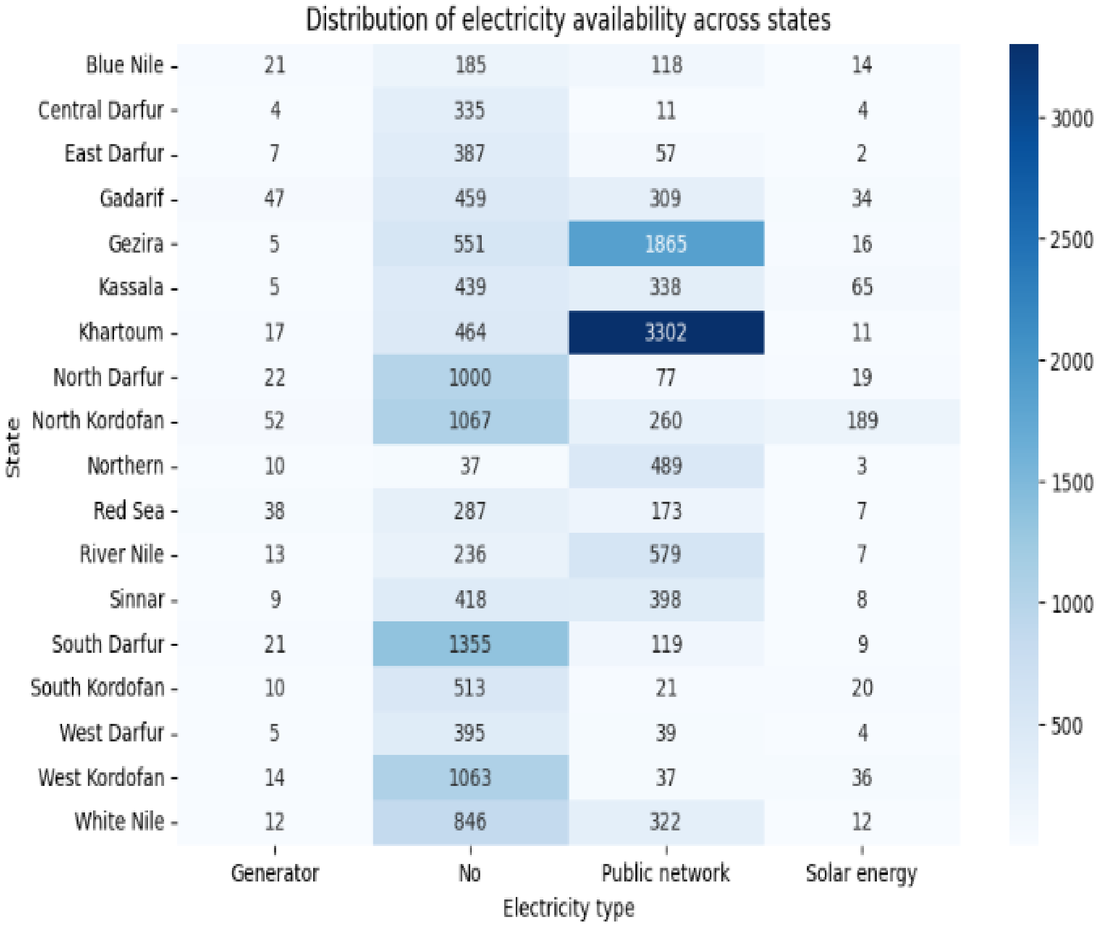


4.7. *Distribution of schools based on their status.*

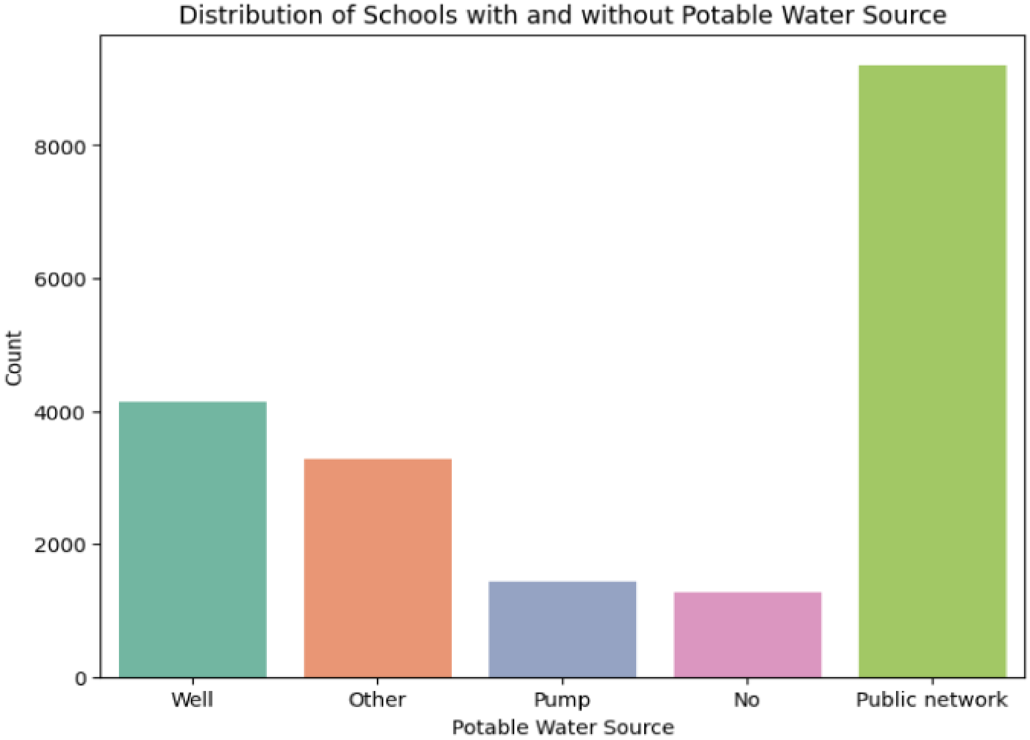
Status	Count	Percentage
normal	14745	76.10%
nongovernmental	3141	16.21%
nomadic	1130	5.83%
special needs	233	1.20%
quranic	72	0.37%
displaced	33	0.17%
complementary	21	0.11%

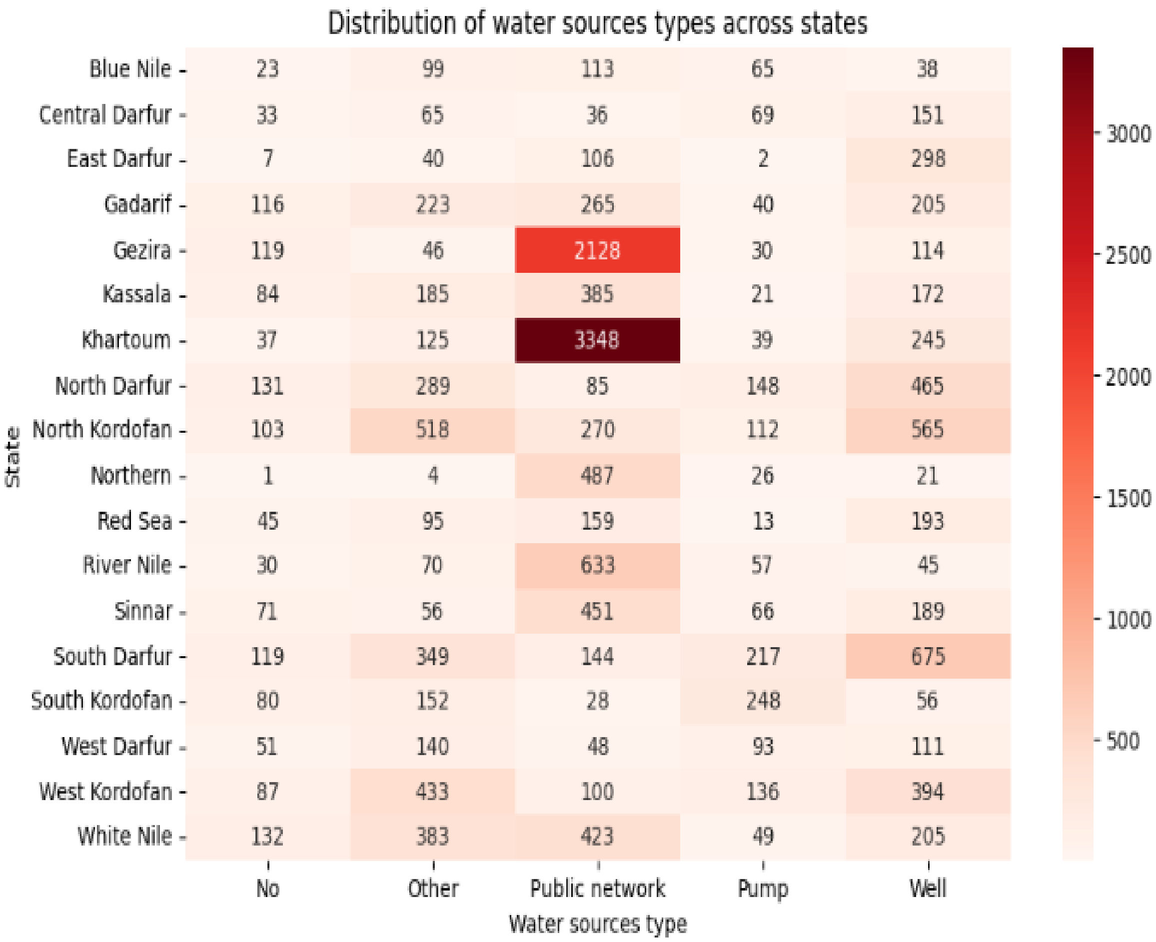
4.8. Distribution of Schools with and without electricity





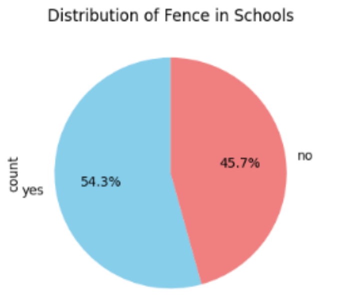
4.9. *Distribution of Schools with and without water resources*

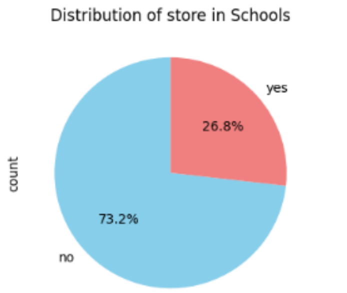


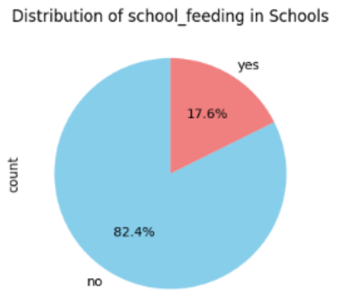


4.10. *Distribution of Schools latrines*

4.11. Distribution of fence, store, school feeding availability







4. Feature Engineering and Data Transformation

```
: Index(['School ID', 'school_name_arabic', 'school_name_english', 'state_name',
       'STCODE', 'locality_name', 'LOCENG', 'LOCCODE', 'location', 'Type',
       'Status', 'teachers', 'students_f', 'students_m', 'students_total',
       'grade1', 'grade2', 'grade3', 'grade4', 'grade5', 'grade6', 'grade7',
       'grade8', 'Total_Classrooms', 'Permanent', 'Needs Rehabilitation',
       'Fence', 'store', 'school_feeding', 'kindergarten', 'kinder_level1',
       'kinder_level2', 'electricity', 'Potable_Water_source', 'Latrines',
       'Latrine_male', 'Latrine_female', 'Latrine_common'],
      dtype='object')
```

- 1st and 5th columns written in Arabic and as both have similar information column written in English ==> Drop
- School name, state name, location (LOCENG) ===> drop
- School grade1 to grade8 columns ==> drop
- edit STCODE and LOCCODE columns to be only the numerical part without "SD" (SD01 – SD18)
- Convert yes/no to 1/0 in columns; Fence, Store, School_feeding, Latrines
- check unique classes in 'location', 'Type', 'Status', 'electricity', 'Potable_Water_source' columns convert them to numerical.

```
Unique classes in location: ['urban' 'rural']
Unique classes in Type: ['Boys' 'Girls' 'Mixed']
Unique classes in Status: ['normal' 'nomadic' 'nongovernmental' 'special needs' 'quranic'
 'complementary' 'displaced']
Unique classes in electricity: ['No' 'Solar energy' 'Generator' 'Public network']
Unique classes in Potable_Water_source: ['Well' 'Other' 'Pump' 'No' 'Public network']
```

- check datatypes

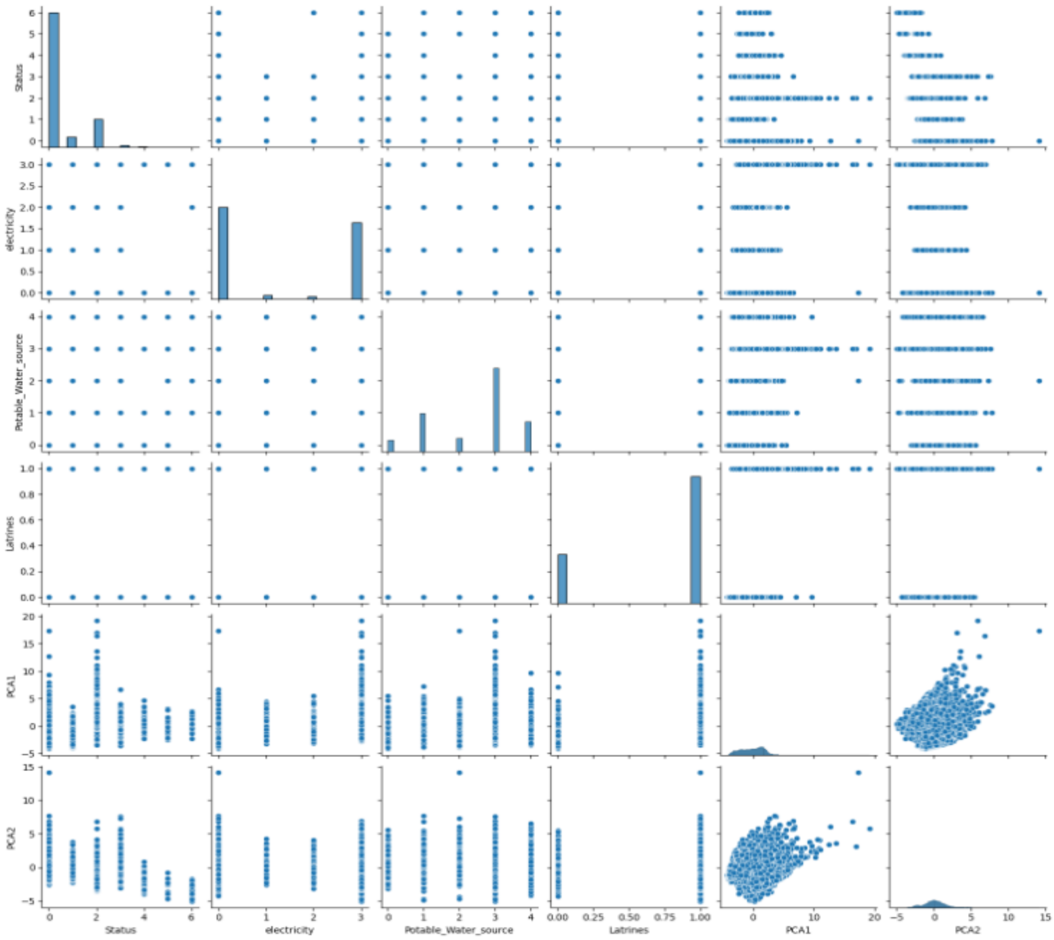
```
School ID                  int64
STCODE                     int64
LOCCODE                    int64
location                   int64
Type                       int64
Status                     int64
teachers                   int64
students_f                 int64
students_m                 int64
students_total             int64
Total_Classrooms         float64
Permanent                float64
Needs Rehabilitation     float64
Fence                      int64
store                      int64
school_feeding             int64
kindergarten               int64
kinder_level1            float64
kinder_level2            float64
electricity                int64
Potable_Water_source       int64
Latrines                   int64
Latrine_male             float64
Latrine_female           float64
Latrine_common           float64
dtype: object
```

- Feature ranking

```

X= df[['School ID', 'STCODE', 'LOCCODE', 'location', 'Type', 'Status',
       'teachers', 'students_f', 'students_m', 'students_total',
       'Total_Classrooms', 'Permanent', 'Needs Rehabilitation', 'Fence',
       'store', 'school_feeding', 'kindergarten', 'kinder_level1',
       'kinder_level2', 'electricity', 'Potable_Water_source', 'Latrines',
       'Latrine_male', 'Latrine_female', 'Latrine_common', 'PCA1',
       'PCA2']].values

y = df['Cluster'].values


rfe = RFE(estimator=DecisionTreeClassifier(), n_features_to_select=6) # define the model
rfe.fit(X, y)# fit RFE
for i in range(X.shape[1]):
    print('Feature: %d, Selected %s, Rank: %.1f' % (i, rfe.support_[i], rfe.ranking_[i]))
```

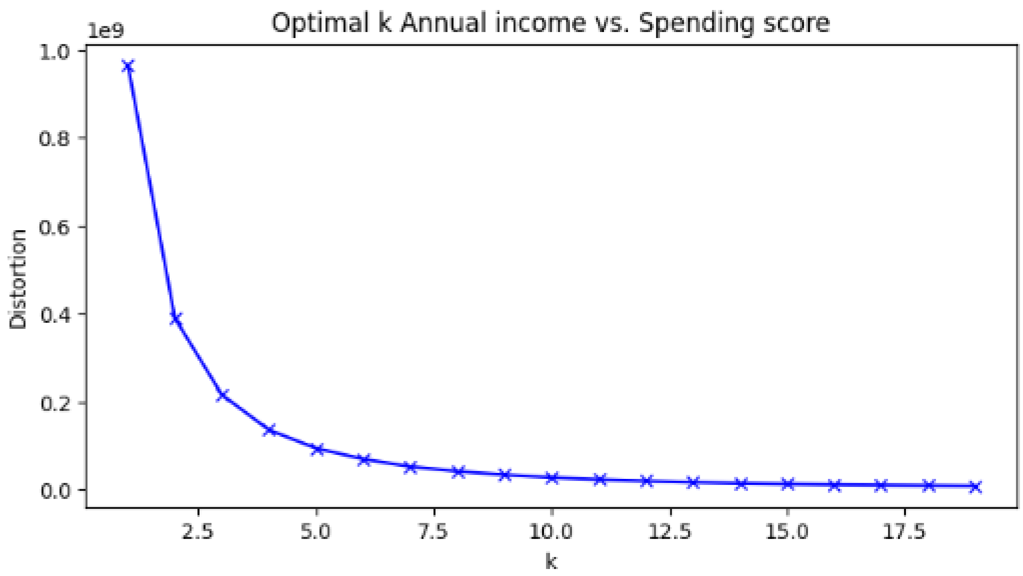
```
Feature: 0, Selected False, Rank: 9.0
Feature: 1, Selected False, Rank: 19.0
Feature: 2, Selected False, Rank: 11.0
Feature: 3, Selected False, Rank: 22.0
Feature: 4, Selected False, Rank: 4.0
Feature: 5, Selected True, Rank: 1.0
Feature: 6, Selected False, Rank: 7.0
Feature: 7, Selected False, Rank: 8.0
Feature: 8, Selected False, Rank: 15.0
Feature: 9, Selected False, Rank: 2.0
Feature: 10, Selected True, Rank: 1.0
Feature: 11, Selected False, Rank: 3.0
Feature: 12, Selected False, Rank: 14.0
Feature: 13, Selected False, Rank: 5.0
Feature: 14, Selected False, Rank: 17.0
Feature: 15, Selected False, Rank: 10.0
Feature: 16, Selected False, Rank: 20.0
Feature: 17, Selected False, Rank: 16.0
Feature: 18, Selected False, Rank: 12.0
Feature: 19, Selected True, Rank: 1.0
Feature: 20, Selected False, Rank: 6.0
Feature: 21, Selected True, Rank: 1.0
Feature: 22, Selected False, Rank: 18.0
Feature: 23, Selected False, Rank: 13.0
Feature: 24, Selected False, Rank: 21.0
Feature: 25, Selected True, Rank: 1.0
Feature: 26, Selected True, Rank: 1.0
```

- Pearson correlation for selected features

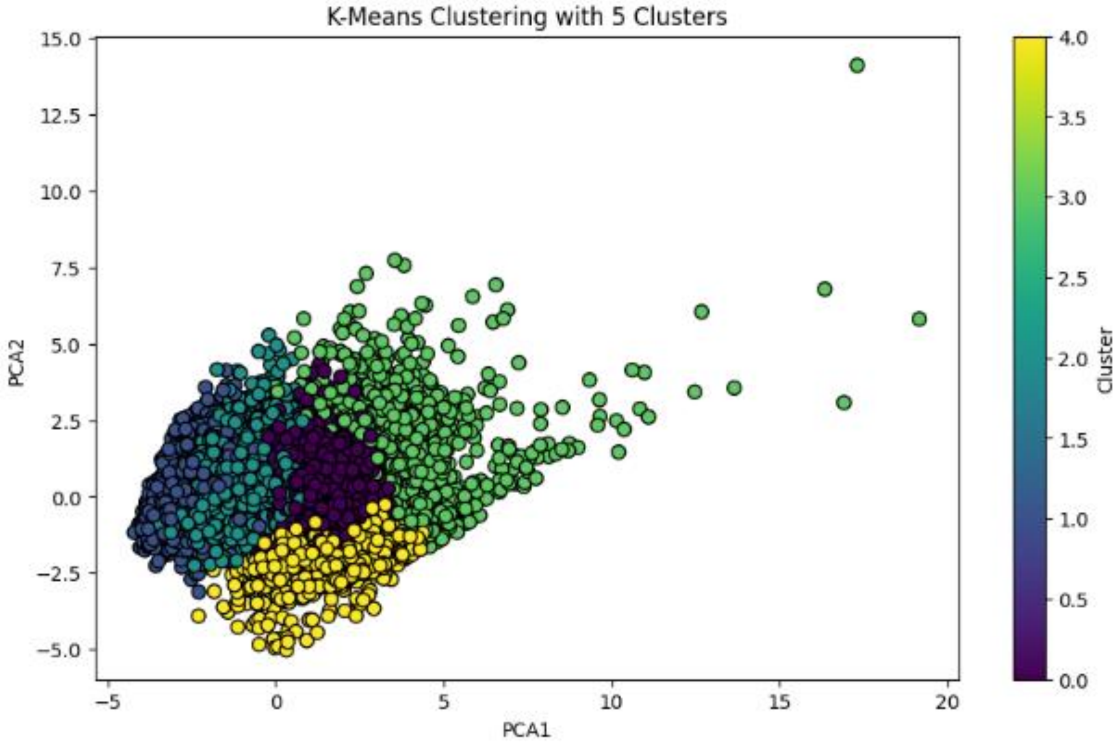


Clustering

- K-means k selecting using “elbow” method



- Cluster into 5 classes



Classification Model Exploration

1. Models, training, evaluation and code implementation

Training set = (14972, 6) (14972,)
Test set = (3744, 6) (3744,)

Model	Metrices
Naïve bayes	Fitting 10 folds for each of 25 candidates, totalling 250 fits Classification report precision recall f1-score support 0 0.90 0.89 0.89 1154 1 0.96 0.96 0.96 929 2 0.89 0.93 0.91 912 3 0.86 0.74 0.80 172 4 0.96 0.97 0.96 577 accuracy 0.92 3744 macro avg 0.91 0.90 0.90 3744 weighted avg 0.92 0.92 0.92 3744 Class Confusion Matrix [[1022 31 93 3 5] [35 892 0 2 0] [45 2 846 9 10] [30 1 5 128 8] [1 5 6 7 558]]
Random forest	Classification report for Random Forest precision recall f1-score support 0 0.94 0.95 0.94 1154 1 0.98 0.98 0.98 929 2 0.96 0.94 0.95 912 3 0.92 0.90 0.91 172 4 0.99 0.99 0.99 577 accuracy 0.96 3744 macro avg 0.96 0.95 0.95 3744 weighted avg 0.96 0.96 0.96 3744 Confusion Matrix for Random Forest: [[1091 13 39 11 0] [14 915 0 0 0] [43 2 861 0 6] [15 0 1 155 1] [3 2 0 3 569]]
KNN	Classification report with k=6 precision recall f1-score support 0 0.92 0.95 0.93 1154 1 0.98 0.97 0.98 929 2 0.95 0.93 0.94 912 3 0.95 0.89 0.92 172 4 0.99 0.98 0.99 577 accuracy 0.95 3744 macro avg 0.96 0.95 0.95 3744 weighted avg 0.95 0.95 0.95 3744 Class Confusion Matrix [[1097 15 36 6 0] [22 905 1 0 1] [60 2 848 0 2] [17 0 0 153 2] [2 2 4 2 567]]

SVM

```
Classification report with linear kernel
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      1154
           1       0.98      0.99      0.98       929
           2       0.95      0.95      0.95       912
           3       0.93      0.93      0.93       172
           4       0.99      0.98      0.99       577

    accuracy                           0.96      3744
   macro avg       0.96      0.96      0.96      3744
weighted avg       0.96      0.96      0.96      3744

Class Confusion Matrix
 [[1087   14   44    9    0]
 [  11  918    0    0    0]
 [  42    2  865    0    3]
 [  10    0    0  160    2]
 [   4    2    2    3  566]]
```

Neural network

Training set = (11977, 27)
(11977, 5)

Validation set = (2995, 27)

(2995, 5)

Test set = (3744, 27)
(3744, 5)

```
Epoch 1/10
375/375 [==============================] - 2s 3ms/step - loss: 0.3425 - accuracy: 0.9026 - val_loss: 0.0992 - val_accuracy: 0.9
659
Epoch 2/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0686 - accuracy: 0.9765 - val_loss: 0.0599 - val_accuracy: 0.9
773
Epoch 3/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0467 - accuracy: 0.9836 - val_loss: 0.0568 - val_accuracy: 0.9
796
Epoch 4/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0389 - accuracy: 0.9869 - val_loss: 0.0429 - val_accuracy: 0.9
816
Epoch 5/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0313 - accuracy: 0.9891 - val_loss: 0.0390 - val_accuracy: 0.9
806
Epoch 6/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0285 - accuracy: 0.9896 - val_loss: 0.0387 - val_accuracy: 0.9
820
Epoch 7/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0253 - accuracy: 0.9926 - val_loss: 0.0691 - val_accuracy: 0.9
820
Epoch 8/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0237 - accuracy: 0.9911 - val_loss: 0.0474 - val_accuracy: 0.9
843
Epoch 9/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0200 - accuracy: 0.9929 - val_loss: 0.0506 - val_accuracy: 0.9
860
Epoch 10/10
375/375 [==============================] - 1s 2ms/step - loss: 0.0185 - accuracy: 0.9938 - val_loss: 0.0308 - val_accuracy: 0.9
856
117/117 [==============================] - 0s 1ms/step - loss: 0.0490 - accuracy: 0.9848
Test accuracy: 0.9847756624221802
```

```
117/117 [==============================] - 0s 1ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      1154
           1       0.99      1.00      0.99       929
           2       0.98      0.99      0.98       912
           3       0.94      0.98      0.96       172
           4       0.99      0.99      0.99       577

    accuracy                           0.98      3744
   macro avg       0.98      0.98      0.98      3744
weighted avg       0.98      0.98      0.98      3744

Confusion Matrix:
[[1121    4   19    9    1]
 [   4  925    0    0    0]
 [   4    3  901    0    4]
 [   3    0    1  168    0]
 [   3    0    0    2  572]]
```