

Data Science Tool Box Python Programming

**PROJECT REPORT**

(Project Semester January-April 2025)

***( Exploring U.S. School Data with Python)***

Submitted by

**(Surendra Reddy)**

Registration No. **12303027**

Programme and Section. Data science K23GN

Course Code . CSE375

Under the Guidance of

**(Aashima )**

**Discipline of CSE/IT**

**Lovely School of Computer Science**

**Lovely Professional University, Phagwara**

**CERTIFICATE**

This is to certify that Surendra Reddy (student’s name) bearing Registration no 12303027 has completed CSE-375 project titled, **“**Project Semester January-April 2025**”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

**DECLARATION**

I Surendra Reddy student of Computer Science under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: Signature

Registration No. 12303027 Name of the student: Surendra Reddy

|  |  |
| --- | --- |
| Title | **Pg .no** |
| **1. Introduction** | **3** |
| **2.** **Source of Dataset** | **4** |
| **3. EDA Process** | **4** |
| **4. Analysis on Dataset** | **4** |
| **4.1 Data Loading and Inspection** | **4** |
| **4.2 Handling Missing Data** | **5** |
| **4.3 Statistical Summary** | **6** |
| **4.4 State-wise School Distribution** | **6** |
| **4.5 Correlation Heatmap** | **9** |
| **4.6 Outlier Detection Using Boxplot** | **10** |
| **4.7 Pair Plot** | **11** |
| **5. Conclusion** | **12** |
| **6. Future Scope** | **12** |
| **7. References** | **12** |
| **8. Source Code** | **13** |

**1. Introduction:**

Education is a fundamental pillar of any thriving society. It plays a crucial role in shaping individuals, building careers, and contributing to the economic development of a nation. In today’s data-driven world, leveraging data analytics in the education sector enables policymakers and institutions to make informed decisions, optimize resources, and enhance learning outcomes.

This report focuses on analyzing the 2016-17 Federal School Code List dataset, providing a comprehensive overview of educational institutions across the United States using Python's data science libraries.

**2.** **Source of Dataset**

The dataset used in this analysis is the Federal School Code List for 2016-2017, provided by the U.S. Department of Education. It contains valuable information about educational institutions including names, locations, school codes, and more.

* File Name: 1617fedschoolcodelist.xls
* Source: https: <https://catalog.data.gov/dataset/national-student-loan-data-system-722b0/resource/be55e2fe-2e34-4667-89e8-a266c456c534>

**3. EDA Process**

The Exploratory Data Analysis (EDA) process includes the following steps:

* Loading and understanding the dataset.
* Handling missing data.
* Generating summary statistics.
* Visualizing insights using graphs and charts.
* Identifying outliers and patterns.

**Python libraries used:**

* pandas for data manipulation
* numpy for numerical operations
* matplotlib and seaborn for visualization

**4. Analysis on Dataset**

**4.1 Data Loading and Inspection**

**i. Introduction**:  
We begin by loading the Excel dataset using pandas and printing basic information.

**ii. General Description**:  
The dataset includes fields such as School Name, State, City, Zip Code, and Federal School Code.

**iii. Code**:

python

CopyEdit

df = pd.read\_excel(""C:\Users\SurendraReddy\Downloads\1617fedschoolcodelist.xls"")

print(df.info())

print(df.head())

**iv. Analysis Results**:

* Number of entries
* Column data types
* First few rows

**v. Visualization**:  
*N/A for this section*

**4.2 Handling Missing Data**

**i. Introduction**:  
Real-world datasets often have missing values. Handling them ensures accurate analysis.

**iii. Code**:

python

CopyEdit

print(df.isnull().sum())

df.fillna(method='ffill', inplace=True)

**iv. Analysis Results**:  
Missing values were filled using forward fill.

**4.3 Statistical Summary**

**i. Introduction**:  
Provides basic descriptive statistics.

**iii. Code**:

python

CopyEdit

print(df.describe(include='all'))

**iv. Analysis Results**:  
Insight into numeric and object-type columns, such as counts, unique values, and mean.

**4.4 State-wise School Distribution**

**i. Introduction**:  
To find which states have the most schools.

**iii. Code**:

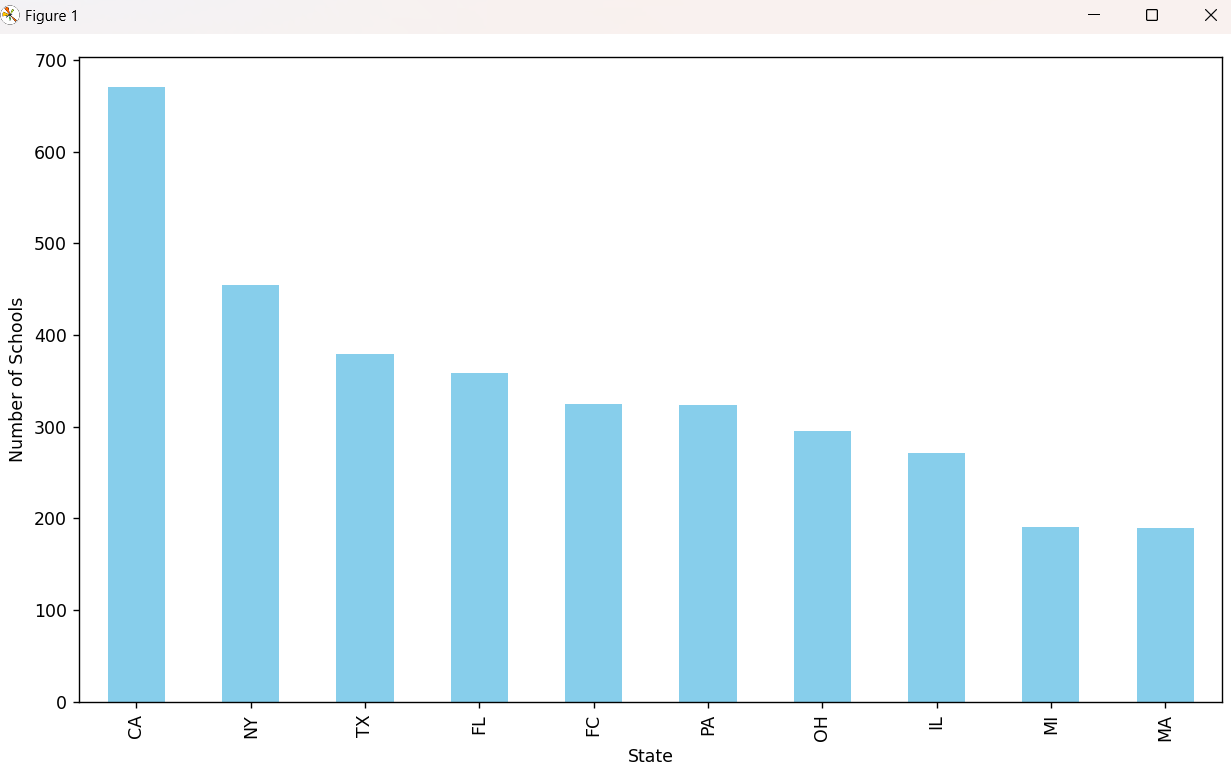
python

CopyEdit

df['StateCode'].value\_counts().head()

**iv. Analysis Results**:  
Top 5 states with the highest number of schools.

**v. Visualization**:  
**Bar Graph - Top 10 States by Number of Schools**  
📍 *[Insert bar Graph Here]*



**Line Graph - School Count by State**  
📍 *[Insert Line Graph Here]*

A graph with a green line

AI-generated content may be incorrect.

**Pie Chart - Top 5 States**  
📍 *[Insert Pie Chart Here]*

A pie chart with different colored circles

AI-generated content may be incorrect.

**4.5 Correlation Heatmap**

**i. Introduction**:  
Helps identify relationships between numerical columns.

**iii. Code**:

python

CopyEdit

numeric\_data = df.select\_dtypes(include=[np.number])

sns.heatmap(numeric\_data.corr(), annot=True, cmap='Blues')

**v. Visualization**:  
📍 *[Insert Heatmap Here]*

A blue rectangle with black text

AI-generated content may be incorrect.

**4.6 Outlier Detection Using Boxplot**

**i. Introduction**:  
Boxplots visualize the spread of data and detect outliers.

**v. Visualization**:  
📍 *[Insert Boxplots Here for each numeric column]*

.0A blue rectangular object with black text

AI-generated content may be incorrect.

**4.7 Pair Plot**

**i. Introduction**:  
Visual comparison of multiple numerical columns.

**v. Visualization**:  
📍 *[Insert Pair Plot Here]*

A screen shot of a scatter plot

AI-generated content may be incorrect.

**5. Conclusion**

Through this exploratory analysis, we obtained significant insights into the distribution and characteristics of schools across the United States. We identified which states have the most institutions and examined patterns in the numeric data.

The EDA helped clean and understand the dataset better, preparing it for further analysis like clustering or prediction.

**6. Future Scope**

* **Predictive Modeling**: Using machine learning to predict school performance.
* **Geographic Mapping**: Visualizing school distribution using GIS.
* **Year-over-Year Comparison**: Analyzing trends across years.
* **Data Enrichment**: Adding socioeconomic or demographic data to analyze school equity.

**7. References**

* U.S. Department of Education: https://www2.ed.gov
* pandas Documentation: https://pandas.pydata.org/
* matplotlib Documentation: <https://matplotlib.org/>
* seaborn Documentation: https://seaborn.pydata.org/
* Python Official: <https://www.python.org/>

**8. Source Code:**

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

# 1. Load the dataset  
df = pd.read\_excel("C:\\Users\\Surendra Reddy\\Downloads\\1617fedschoolcodelist.xls")

# 2. Get info about the dataset  
print(df.info())  
print("\nFirst 5 Rows:")  
print(df.head())

# 3. Handling missing data  
print("\nMissing Values in Each Column:")  
print(df.isnull().sum())

# Fill missing values with forward fill (basic way)  
df.fillna(method='ffill', inplace=True)

# 4. Statistical Summary  
print("\nSummary Statistics:")  
print(df.describe(include='all'))

# 5. Number of Schools by State  
print("\nTop 5 States by Number of Schools:")  
print(df['StateCode'].value\_counts().head())

# Bar Graph - Top 10 States by Number of Schools  
plt.figure(figsize=(10, 6))  
df['StateCode'].value\_counts().head(10).plot(kind='bar', color='skyblue')  
plt.xlabel("State")  
plt.ylabel("Number of Schools")  
plt.tight\_layout()  
plt.show()

# Line Graph - Simulated School Count by State  
state\_counts = df['StateCode'].value\_counts().head(10)  
plt.plot(state\_counts.index, state\_counts.values, marker='o', color='green')  
plt.title("School Count by State")  
plt.xlabel("State")  
plt.ylabel("Count")  
plt.grid(True)  
plt.show()

# Pie Chart - Top 5 States  
top\_states = df['StateCode'].value\_counts().head(5)  
plt.pie(top\_states.values, labels=top\_states.index, autopct='%1.1f%%')  
plt.title("Top 5 States - Pie Chart")  
plt.show()

# 6. Heatmap (only for numeric columns)  
numeric\_data = df.select\_dtypes(include=[np.number])

if not numeric\_data.empty:  
 plt.figure(figsize=(8, 6))  
 sns.heatmap(numeric\_data.corr(), annot=True, cmap='Blues')  
 plt.title("Heatmap of Numeric Data")  
 plt.show()

# 7. Boxplot - Outliers  
for column in numeric\_data.columns:  
 plt.figure()  
 sns.boxplot(y=df[column])  
 plt.title(f"Boxplot of {column}")  
 plt.show()

# Scatter Plot - ZipCode vs Random values (for demo)  
if 'ZipCode' in df.columns:  
 plt.figure(figsize=(8, 5))  
 plt.scatter(df['ZipCode'], np.random.rand(len(df)), alpha=0.5)  
 plt.title("Scatter Plot: ZipCode vs Random")  
 plt.xlabel("ZipCode")  
 plt.ylabel("Random Values")  
 plt.show()

# Pair Plot - if enough numeric data  
if len(numeric\_data.columns) >= 2:  
 sns.pairplot(numeric\_data)  
 plt.suptitle("Pair Plot", y=1.02)  
 plt.show()