School Immunizations

Group Members

Muhammad Adil Inam	19030001
Muhammad Mudasir Yasin Mughal	19030032
Ghulam Mohi Ud Din	19030051
Waqar Ul Haq Khatana	22100199

Instructor
Dr. Asim Karim

DEPARTMENT OF COMPUTER SCIENCE, SYED BABAR ALI
SCHOOL OF SCIENCE AND ENGINEERING
LAHORE UNIVERSITY OF MANAGEMENT SCIENCES

List of Figures

Table of Contents

Chapter 1 Introduction

Overview

The dataset contains the immunization status of kindergarten students in schools of California state from the year 2016 to 2019. Young children are vaccinated to help them prevent the diseases like polio; measles, mumps and rubella (MMR); chickenpox (Varicella); influenza; diphtheria, tetanus and whooping cough (DPT); hepatitis B (HEPB) and much more. The students should be vaccinated and each school should report the vaccination status yearly to the California Department of Public Health.

Dataset Description

The dataset is described by the following parameters

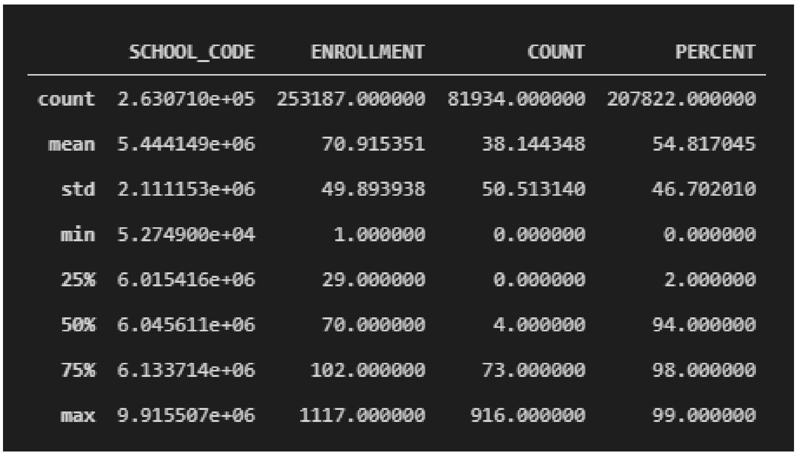
Data Column	Description
Year	It is the year in which immunization was recorded
School code	It is the unique identifier of each school
Country	It is the country in which the school lies
School Sector	It represents that whether the school is of private or of public sector
City	It is the name of the city in which school lies
School name	It is the name of the school
Reported	It indicates whether the school reported the immunization status that year to CDPH[1]
Enrolment	The number of students enrolled that year
Category	It indicates the type of vaccination received
Count	It indicates the number of students vaccinated
Percent	It indicates the percentage of the students vaccinated

[1] California Department of Public Health

Chapter 2 Data Pre-Processing

2.1 Understanding the data

The dataset consists of 8863 different schools from 58 countries of California state. 253187 students were enrolled in these schools during the period of four years from 2016 to 2019. Out of these 253187 students only 81934 were vaccinated. The remaining 171253 students were either not vaccinated or their records are missing from the dataset. There was a total of 11 types of vaccines received by the students.



	SCHOOL_CODE	ENROLLMENT	COUNT	PERCENT
count	2.630710e+05	253187.000000	81934.000000	207822.000000
mean	5.444149e+06	70.915351	38.144348	54.817045
std	2.111153e+06	49.893938	50.513140	46.702010
min	5.274900e+04	1.000000	0.000000	0.000000
25%	6.015416e+06	29.000000	0.000000	2.000000
50%	6.045611e+06	70.000000	4.000000	94.000000
75%	6.133714e+06	102.000000	73.000000	98.000000
max	9.915507e+06	1117.000000	916.000000	99.000000

Figure 1 Summary of the raw dataset

2.2 Organizing the data

The uninteresting or useless data was removed and the dataset was formulated to fulfil our needs.

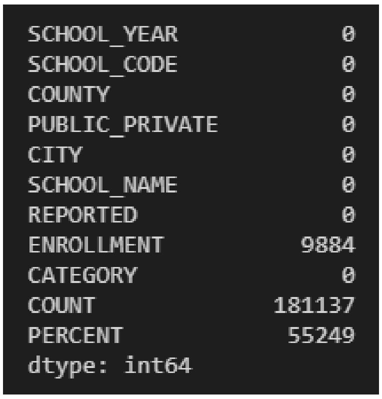
2.2.1 Removing Duplicates

A total of 32 duplicate entries were found and were removed from the dataset. The columns school year and category of immunization were used to determine the duplicates.

```python
def duplicate_rows_based_on_col(df, columns):
	return df[df.duplicated(columns)]
duplicate_rows_based_on_col(data,['SCHOOL_YEAR','CATEGORY'])
```

2.2.2 Countering the Null values

The NAN entries were found in the entire dataset and the number of null values by each column are given below in **Error! Reference source not found.**



```
SCHOOL_YEAR            0
SCHOOL_CODE            0
COUNTY                 0
PUBLIC_PRIVATE         0
CITY                   0
SCHOOL_NAME            0
REPORTED               0
ENROLLMENT          9884
CATEGORY               0
COUNT             181137
PERCENT            55249
dtype: int64
```

Figure 2 Number of NAN entries in each column

Firstly, all the records were dropped where all the three values ENROLLMENT, COUNT and PERCENT were NAN. This left 171253 NAN entries in the COUNT column and 45365 NAN entries in the PERCENT column.

```
df_new = data.dropna(subset=['ENROLLMENT', 'COUNT', 'PERCENT'], thresh=1)
```

Now, the missing values of COUNT were filled using the PERCENT and the ENROLLMENT values. The following code was used to accomplish this task.

```
for index, row in df_new.iterrows():
  if pd.isnull(row['COUNT']) and pd.notnull(row['PERCENT']):
        new_count_value = np.ceil(row['ENROLLMENT'] * (row['PERCENT'] / 100))
        df_new.at[index,'COUNT'] = new_count_value
```

After the above-mentioned steps, 45365 NaN values remained in the COUNT and PERCENT columns. The remaining NaN values were filled by using the average number of enrolments in that year.

```
for row in df_new['SCHOOL_YEAR'].unique():
    df_new_set_by_year = df_new.loc[df_new['SCHOOL_YEAR'] == row]
    for row2 in df_new_set_by_year['SCHOOL_NAME'].unique():
        df_new_set_with_year_and_school_name = df_new_set_by_year[df_new_set_by
        _year['SCHOOL_NAME'] == row2]
        mean_school_year_with_school = np.ceil(df_new_set_with_year_and_school_
name['ENROLLMENT'].mean())
        #where count is nan put the avg
        for index, rows in df_new_set_with_year_and_school_name.iterrows():
            if pd.isnull(rows['COUNT']):
                df_new.at[index,'COUNT'] = mean_school_year_with_school
```

The COUNT values were replaced by the ENROLLMENT values where the COUNT value was greater than the ENROLLMENT after filling the NaN values. It was done as the number of vaccinated students can not be more than the enrolled students.

```
for index, rows in df_new.loc[df_new['ENROLLMENT'] < df_new['COUNT']].iterrows():
    df_new.at[index, 'COUNT'] = rows['ENROLLMENT']
```

The PERCENT column was dropped as it is redundant now and can be calculated from COUNT and ENROLLMENT.

2.2.3 Normalizing the Data

The COUNT and ENROLLMENTS columns were normalized by using the min-max normalizing technique so that the data may be in the same range and to remove any possible bias due to difference in values.

```
normalized_count = []
min_count = df_new['COUNT'].min()
max_count = df_new['COUNT'].max()
for value in df_new['COUNT']:
    normalized_count.append((value - min_count)/(max_count - min_count))

normalized_enrollment = []
min_enrol = df_new['ENROLLMENT'].min()
max_enrol = df_new['ENROLLMENT'].max()
for value in df_new['ENROLLMENT']:
    normalized_enrollment.append((value - min_enrol)/(max_enrol - min_enrol))
```

Chapter 3 Visualization

3.1 Grouping by School Category

First of all, we grouped the dataset by the column SCHOOL_CATEGORY. This grouped data was then visually represented by using a pie chart to represent the percentage of both public and private schools.

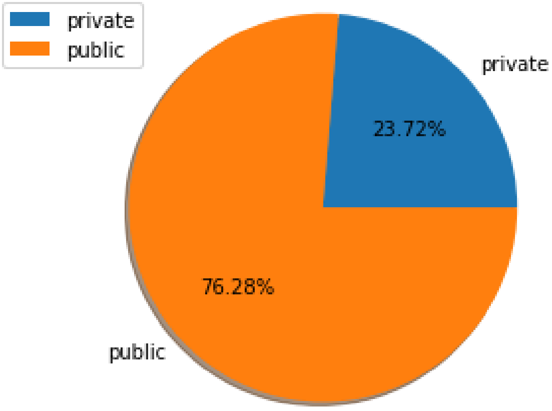


Figure 3 Pie chart representing the percentage of schools by category

3.2 Annual enrollment vs vaccinated students

A vertical grouped bar chart was plotted to show the number of students enrolled each year and the number of students that received vaccinations that year as shown in Figure 4. The enrollment dropped during the year 2017-2018 however almost the same number of students were vaccinated that year.

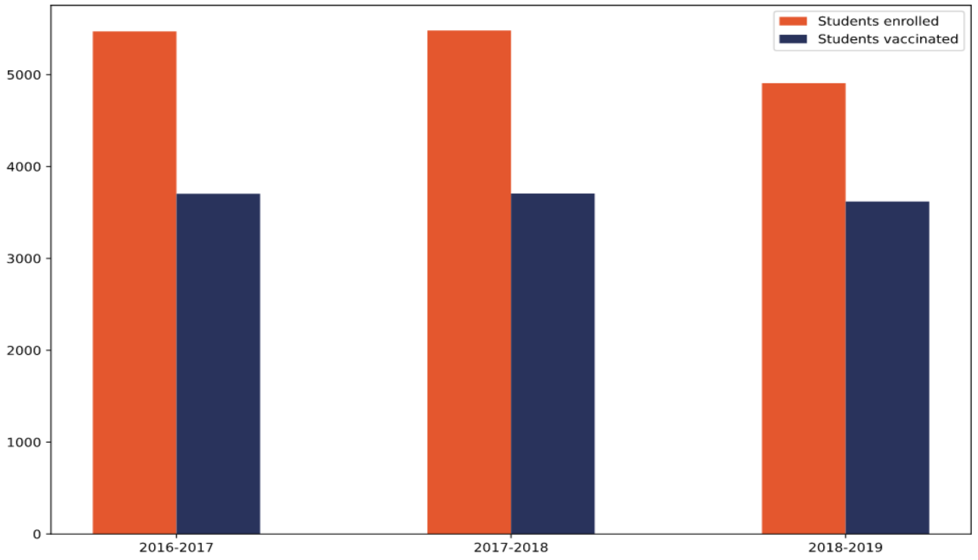


Figure 4 Student's enrollment vs number of students vaccinated annually

3.3 Annual immunization by category

The number of immunizations for each immunization category is shown in the grouped bar chart in Figure 5. The immunization by category dropped in the year 2017-2018 and increased a bit again in the year 2018-2019. In the year 2018-2019, no student received the pbe vaccination.

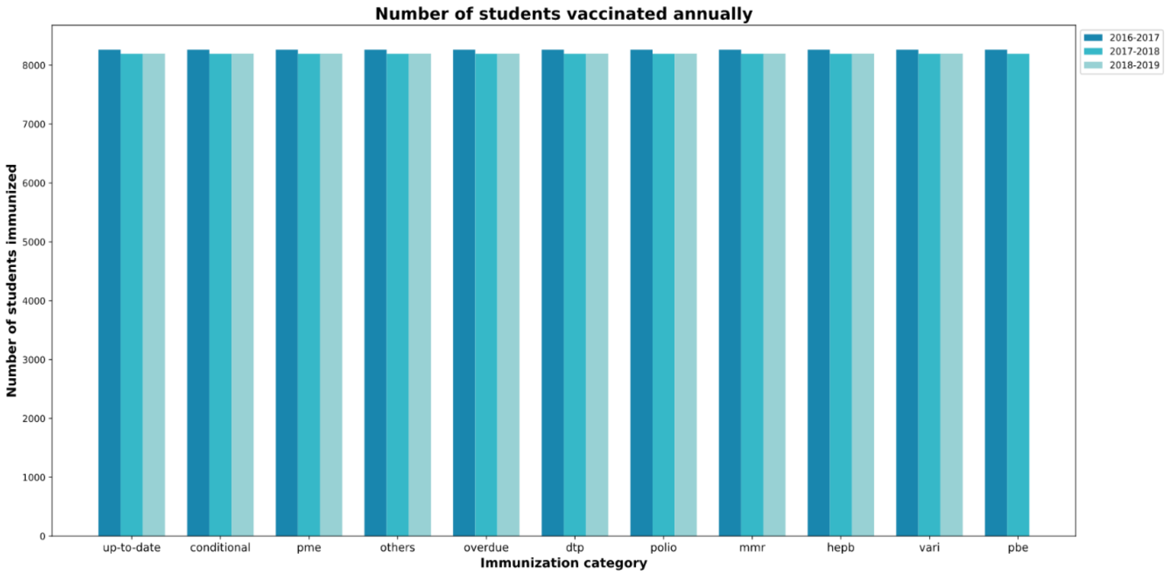


Figure 5 Number of immunizations by category annually

3.3.1 Immunization by school year and category

The bar graph shown in Figure 6 represents the number of immunizations according to the school category annually.

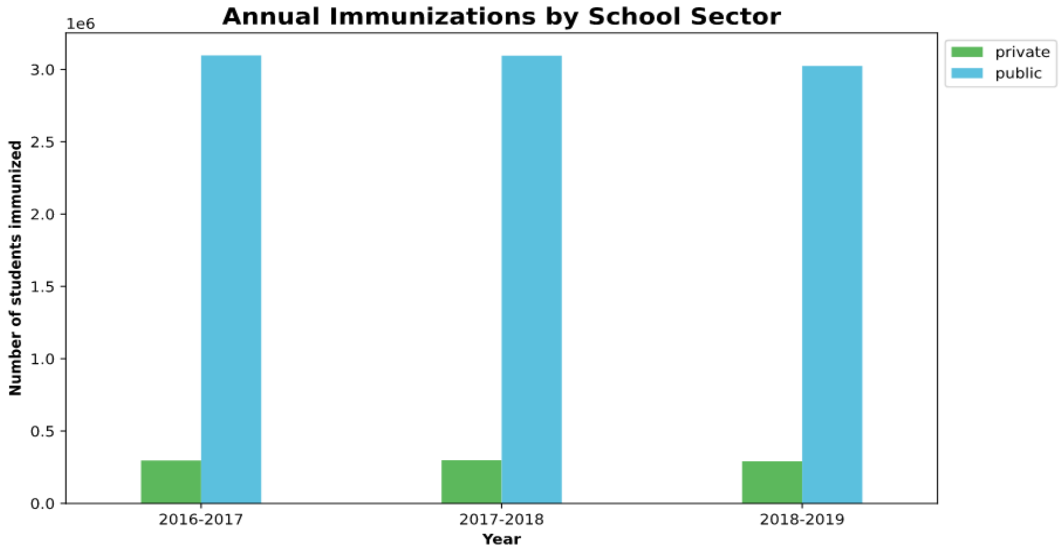


Figure 6 Immunizations done per school category each year

The bar graph shown in Figure 7 represents the number of students immunized by each category of the vaccine in the whole dataset according to the category of the school.

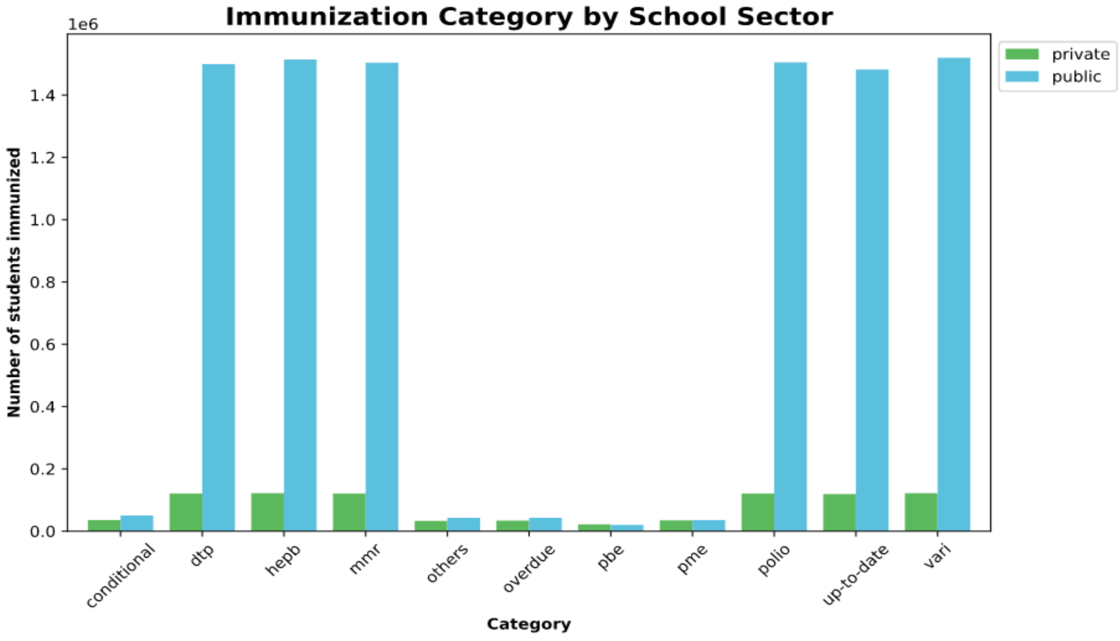


Figure 7 Overall immunizations of each type done per school category

Figure 8 shows the relative percentage of the students who got a vaccine. The data shows that every other vaccine was given to about 9.38% of students while the ***pbe*** vaccine was received by only 6.23% of students.

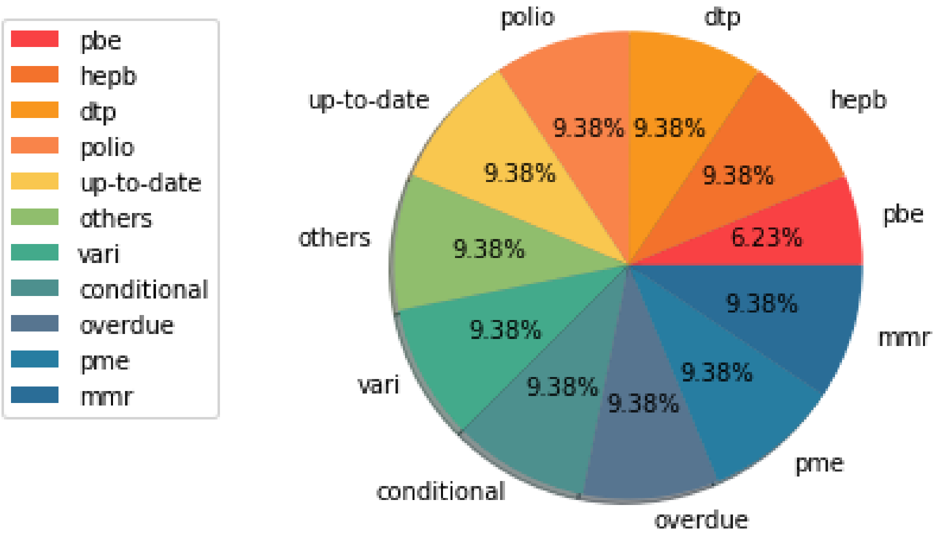


Figure 8 Percentage of students who received a particular vaccine

Below is the boxplot of the dataset variables ***CATEGORY*** and ***COUNT*** in Figure 9**Error! Reference source not found.**.

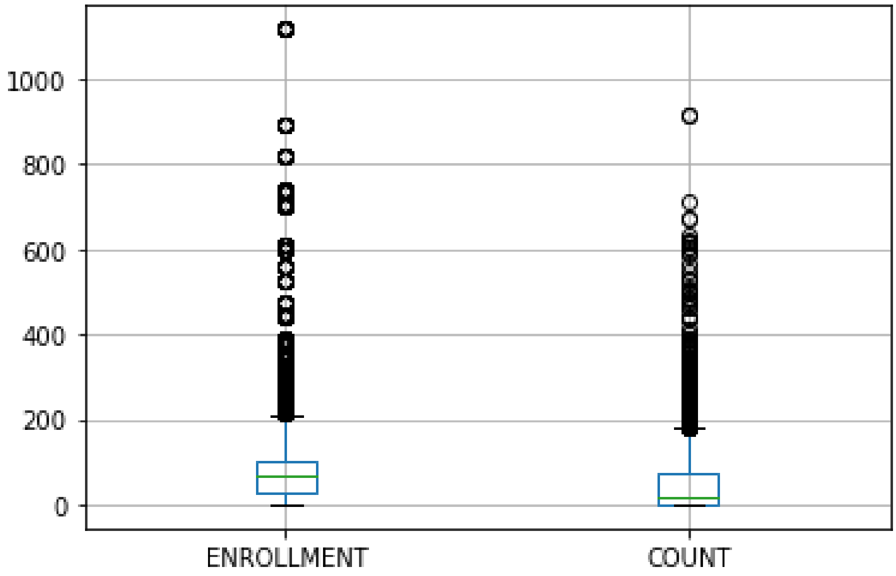


Figure 9 Boxplot of Enrollment and the number of students vaccinated in the whole dataset

The pie chart shown in Figure 10 represents the percentage of the students enrolled per year from the period of 2016 to 2019. The enrolment dropped during the year 2018-2019.

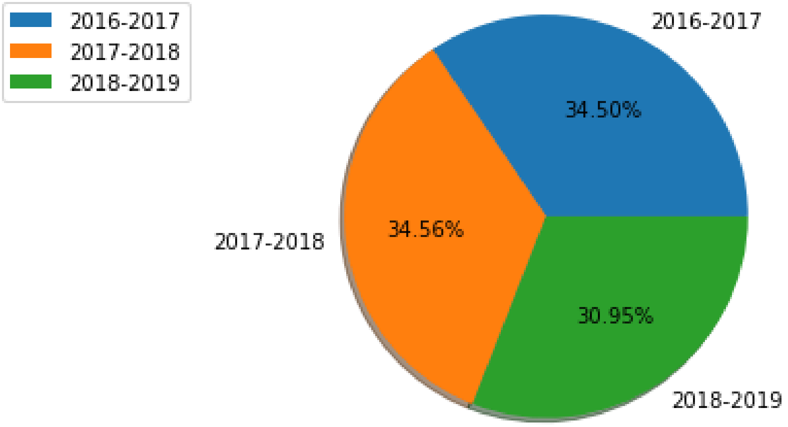


Figure 10 Percentage of students enrolled annually from the year 2016 to 2019

The pie chart shown in Figure 11 represents the percentage of the students immunized annually from the period of 2016 t0 2019. The percentage of students immunized during the year 2018-2019 dropped a bit from the previous two years.

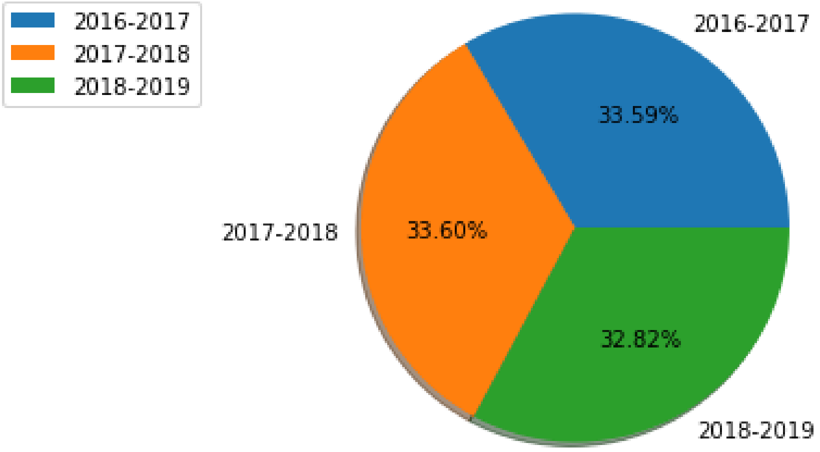


Figure 11 Percentage of students vaccinated annually from year 2016 to 2019

3.4 Correlation between variables

The correlation found between different variables of the whole dataset was plotted as shown in Figure 12. There was a high correlation between the enrolment of students and the category of the school i.e. private or public. Also, the number of students enrolled and the number of vaccinated students is also highly correlated. The school category and the school code are negatively correlated with each other.

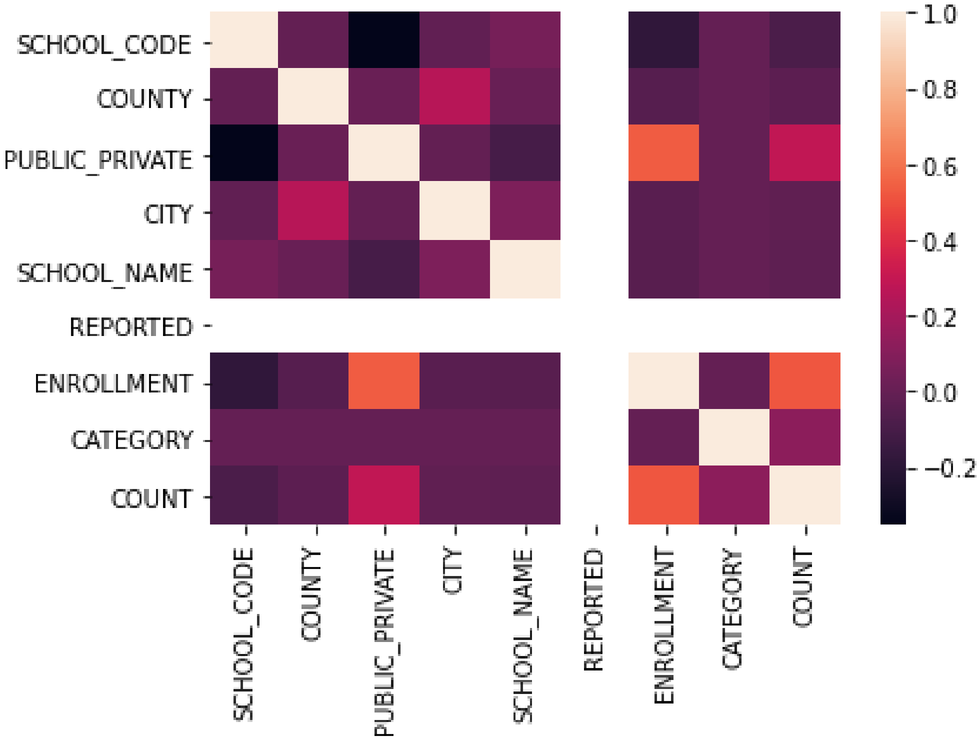
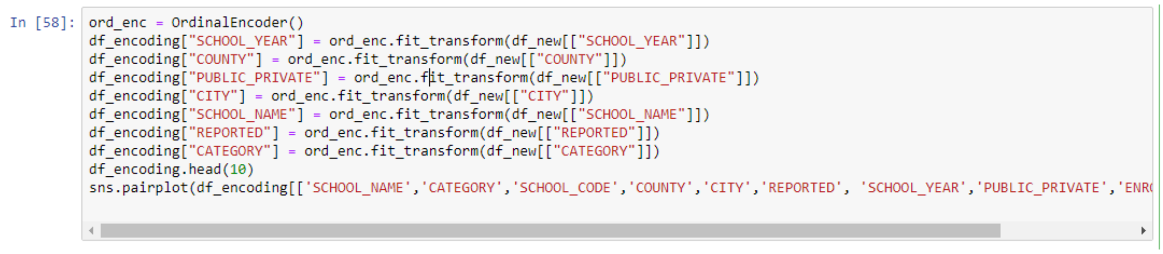


Figure 12 Correlation between different variables

Chapter 4 Analysis

Clustering

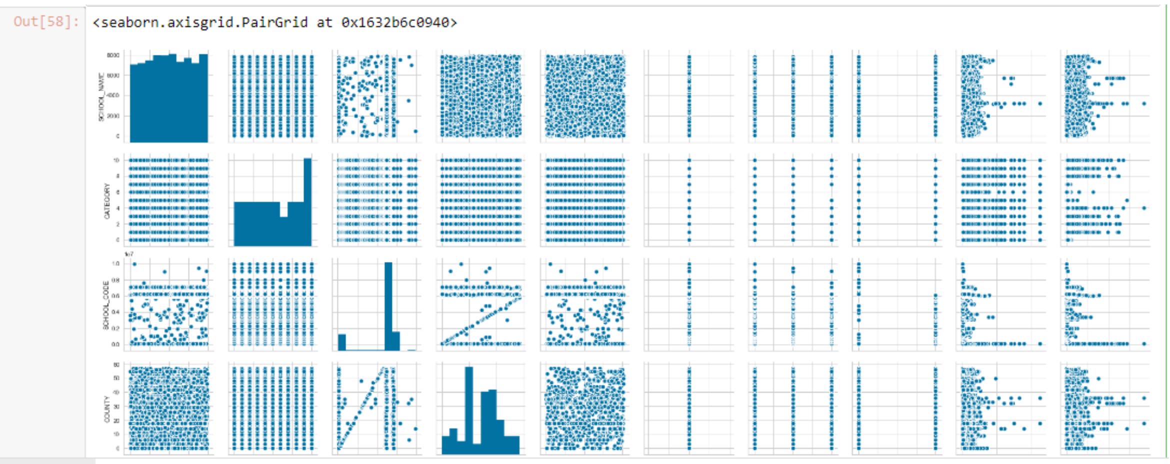
After data preprocessing and visualization, our next step included clustering. First, we used *OrdinalEncoder* to encode all categorical features in our dataset as integer arrays. This gives us the correlation between different features and an idea of which attributes can be used for clustering.



```
In [58]: ord_enc = OrdinalEncoder()
         df_encoding["SCHOOL_YEAR"] = ord_enc.fit_transform(df_new[["SCHOOL_YEAR"]])
         df_encoding["COUNTY"] = ord_enc.fit_transform(df_new[["COUNTY"]])
         df_encoding["PUBLIC_PRIVATE"] = ord_enc.fit_transform(df_new[["PUBLIC_PRIVATE"]])
         df_encoding["CITY"] = ord_enc.fit_transform(df_new[["CITY"]])
         df_encoding["SCHOOL_NAME"] = ord_enc.fit_transform(df_new[["SCHOOL_NAME"]])
         df_encoding["REPORTED"] = ord_enc.fit_transform(df_new[["REPORTED"]])
         df_encoding["CATEGORY"] = ord_enc.fit_transform(df_new[["CATEGORY"]])
         df_encoding.head(10)
         sns.pairplot(df_encoding[['SCHOOL_NAME','CATEGORY','SCHOOL_CODE','COUNTY','CITY','REPORTED', 'SCHOOL_YEAR','PUBLIC_PRIVATE','ENR
```

The correlation found between different variables of the whole dataset was plotted as shown in Figure 13. It gives us a clear idea of which attributes can be used for clustering. From the figure given below, we decided to do clustering based on the following relations.

- County & Category
- School_Name to Category
- Public_Private to Category
- City to Category
- School to Enrolment
- Year to Category



```
Out[58]: <seaborn.axisgrid.PairGrid at 0x1632b6c0940>
```

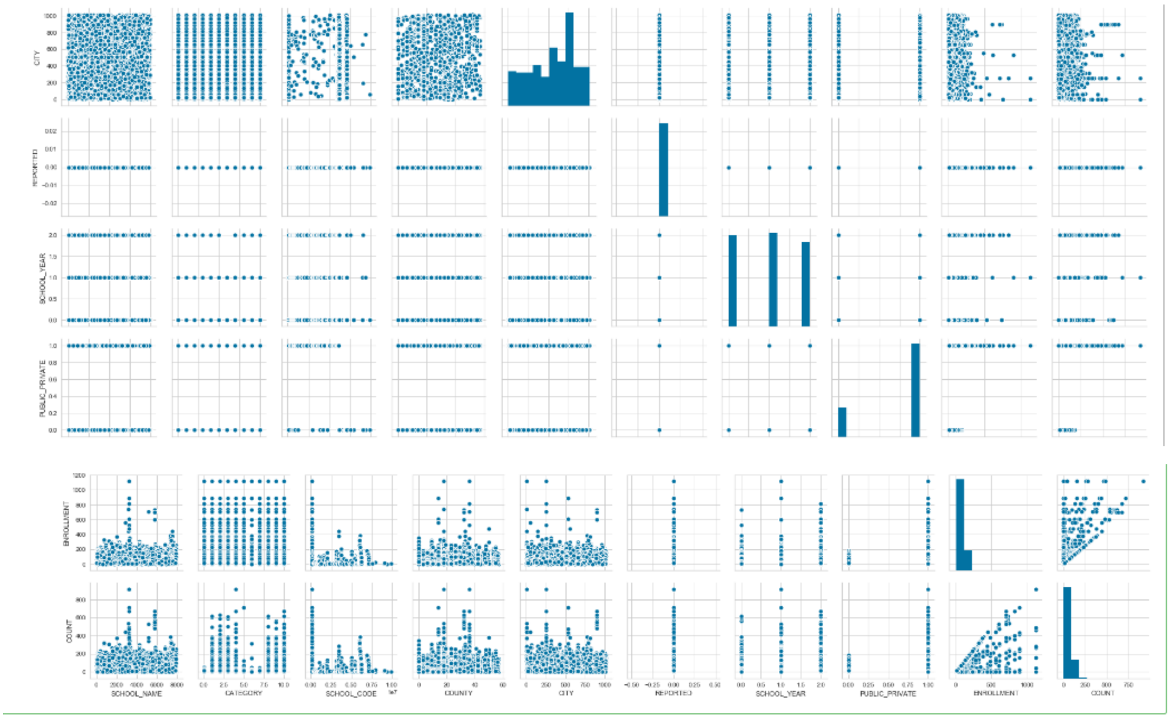


Figure 13 Pair plot of different variables

Then, we used the elbow curve to determine the number of clusters. Elbow curve gives us the optimal number of clusters such that adding another cluster does not give much better modeling of the data.

We used K-Means Clustering and Minibatch K-Means Clustering in this project.

Cluster # 1: County & Category

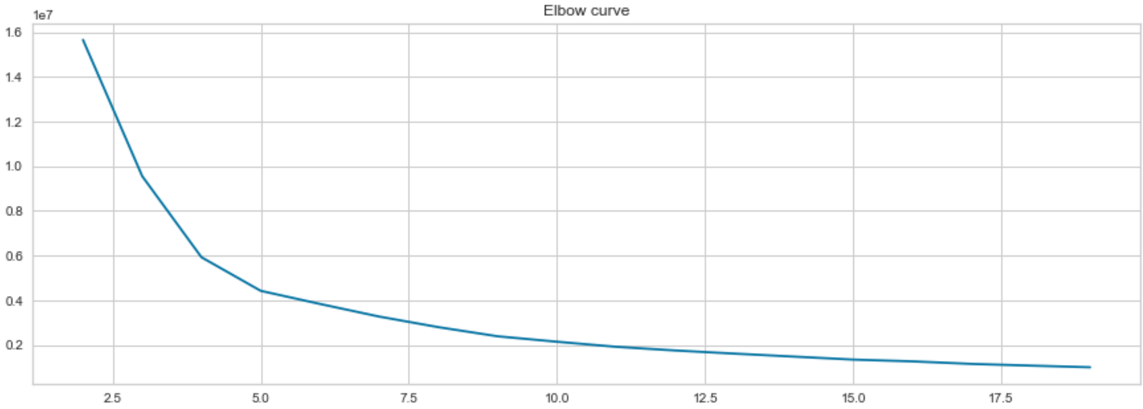


Figure 14 Elbow Curve for County & Category

As can be seen from the graph above, the optimal number of clusters is 5 for County-to-Country mapping. Figures 15 and 16 represent the K-Means and Minibatch K-Means Clustering output for the County-to-Country mapping.

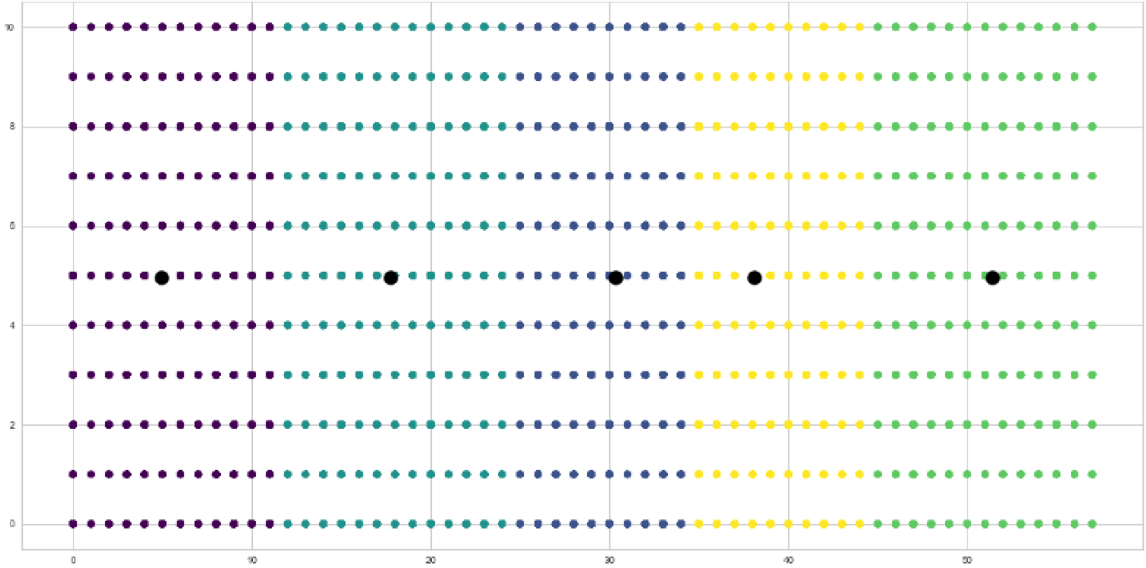


Figure 15 K-Means Clustering for County & Category

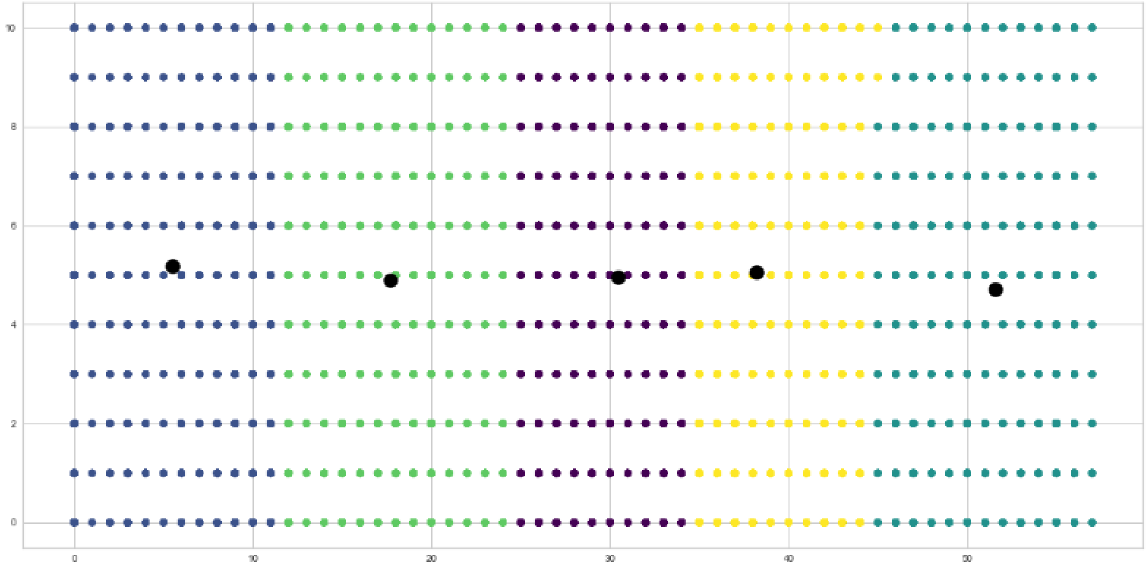


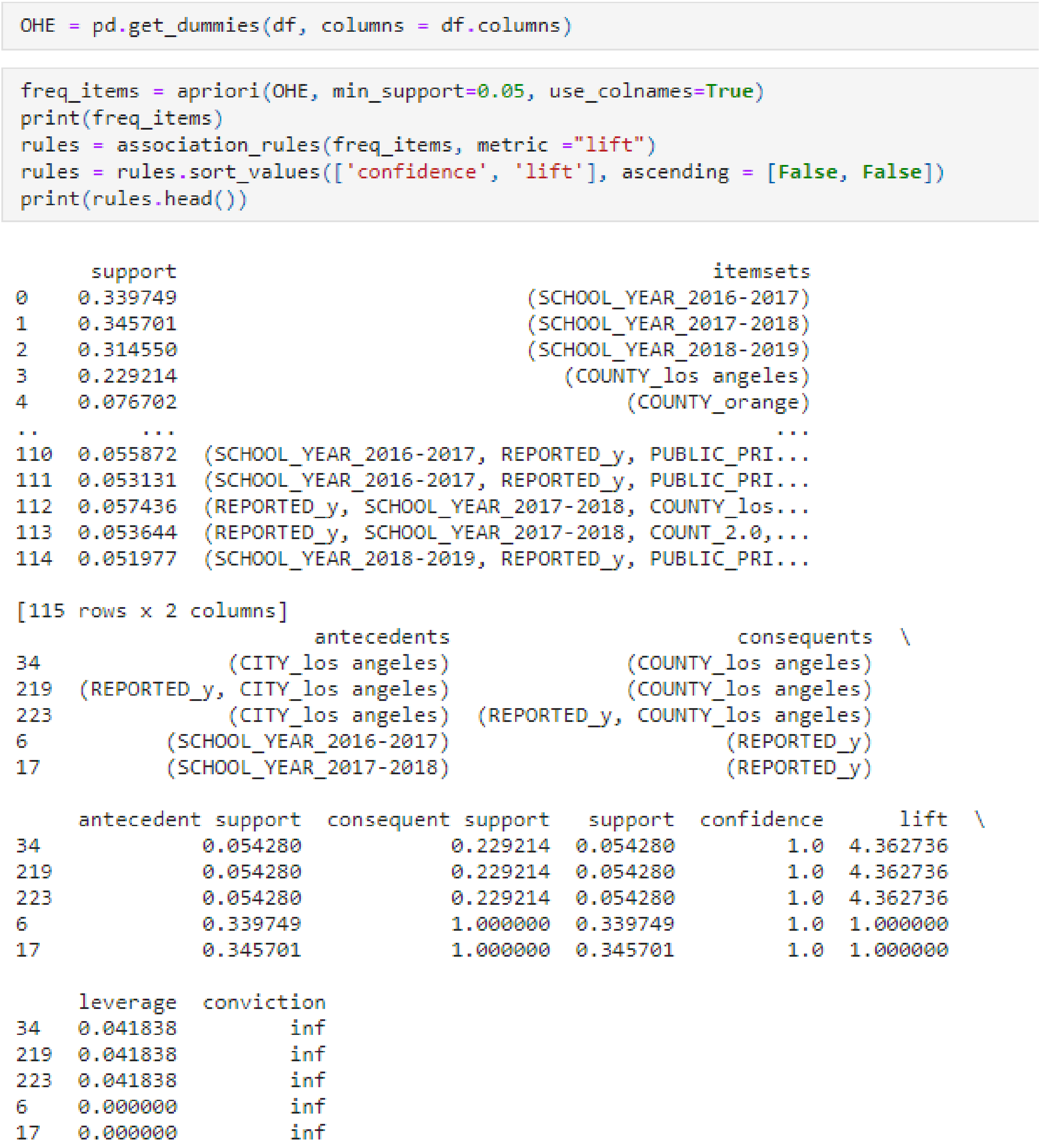
Figure 16 Minibatch K-Means Clustering for County & Category

The remaining cluster analysis for School_Name to Category, Public_Private to Category, City to Category, School to Enrollment, and Year to Category can be found in the code attached.

Frequent Pattern Mining

Next, we used Frequent Pattern Mining to find frequent patterns, associations, or causal structures from the given dataset. This process will help us find the rules that will enable us to predict the occurrence of a specific feature value based on the occurrence of other feature values in the given records.

We used OneHotEncoding on the Category and School_Year attributes first and found frequent patterns using Apriori Algoritm. Then, we applied the same mechanism on the whole database and generated frequents itemsets.



```
OHE = pd.get_dummies(df, columns = df.columns)
```

```
freq_items = apriori(OHE, min_support=0.05, use_colnames=True)
print(freq_items)
rules = association_rules(freq_items, metric ="lift")
rules = rules.sort_values(['confidence', 'lift'], ascending = [False, False])
print(rules.head())
```

```
      support                                       itemsets
0    0.339749                         (SCHOOL_YEAR_2016-2017)
1    0.345701                         (SCHOOL_YEAR_2017-2018)
2    0.314550                         (SCHOOL_YEAR_2018-2019)
3    0.229214                            (COUNTY_los angeles)
4    0.076702                                 (COUNTY_orange)
..        ...                                            ...
110  0.055872  (SCHOOL_YEAR_2016-2017, REPORTED_y, PUBLIC_PRI...
111  0.053131  (SCHOOL_YEAR_2016-2017, REPORTED_y, PUBLIC_PRI...
112  0.057436  (REPORTED_y, SCHOOL_YEAR_2017-2018, COUNTY_los...
113  0.053644  (REPORTED_y, SCHOOL_YEAR_2017-2018, COUNT_2.0,...
114  0.051977  (SCHOOL_YEAR_2018-2019, REPORTED_y, PUBLIC_PRI...

[115 rows x 2 columns]
                       antecedents                       consequents  \
34                 (CITY_los angeles)              (COUNTY_los angeles)
219    (REPORTED_y, CITY_los angeles)              (COUNTY_los angeles)
223                (CITY_los angeles)  (REPORTED_y, COUNTY_los angeles)
6             (SCHOOL_YEAR_2016-2017)                      (REPORTED_y)
17            (SCHOOL_YEAR_2017-2018)                      (REPORTED_y)

     antecedent support  consequent support   support  confidence      lift  \
34             0.054280            0.229214  0.054280         1.0  4.362736
219            0.054280            0.229214  0.054280         1.0  4.362736
223            0.054280            0.229214  0.054280         1.0  4.362736
6              0.339749            1.000000  0.339749         1.0  1.000000
17             0.345701            1.000000  0.345701         1.0  1.000000

     leverage  conviction
34   0.041838         inf
219  0.041838         inf
223  0.041838         inf
6    0.000000         inf
17   0.000000         inf
```

Figure 17 Frequent Itemsets generated using Apriori Algorithm

Chapter 5 Conclusion and Recommendations

The data contains demographic and immunization status of kindergarten students in schools of California state from the year 2016 to 2019. The annual assessment measures the immunization coverage amongst the students entering kindergarten. The data was obtained from California Department of Public Health to which each school submits a yearly vaccination status report.

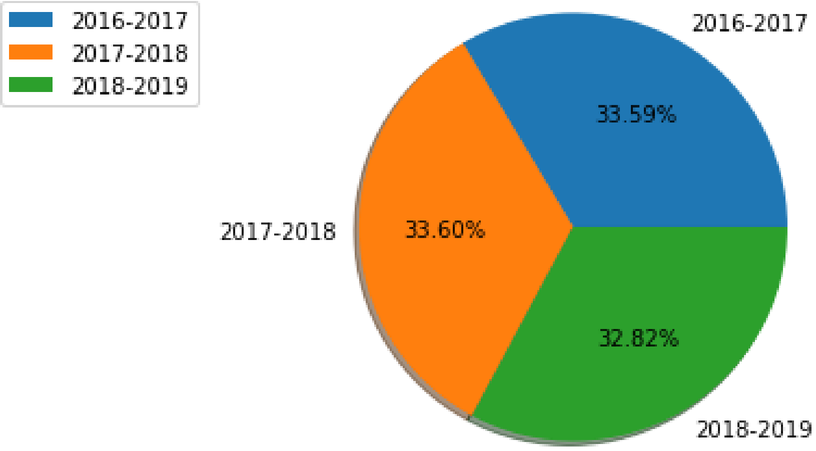


Figure 18 Percentage of students vaccinated annually from year 2016 to 2019

The pie chart shown in Figure 11 represents the percentage of the students immunized annually from the period of 2016 t0 2019. As can be seen, the percentage of students immunized during the year 2018-2019 has decreased compared to the previous two years so the government needs to ensure proper vaccinations for all schools and the relevant protocols should be made more stringent.

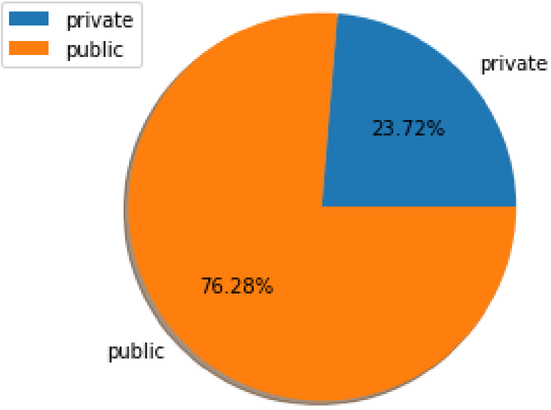


Figure 19 Pie chart representing the percentage of schools by category

The above pie chart represents the percentage of both public and private schools. As shown below almost three quarters of the schools belong the public sector which gives a guideline for the government and concerned authorities to focus more on the immunization of public sector schools. Keeping that in view, the resources allocated for the public sector should ideally be three times that of the private sector.

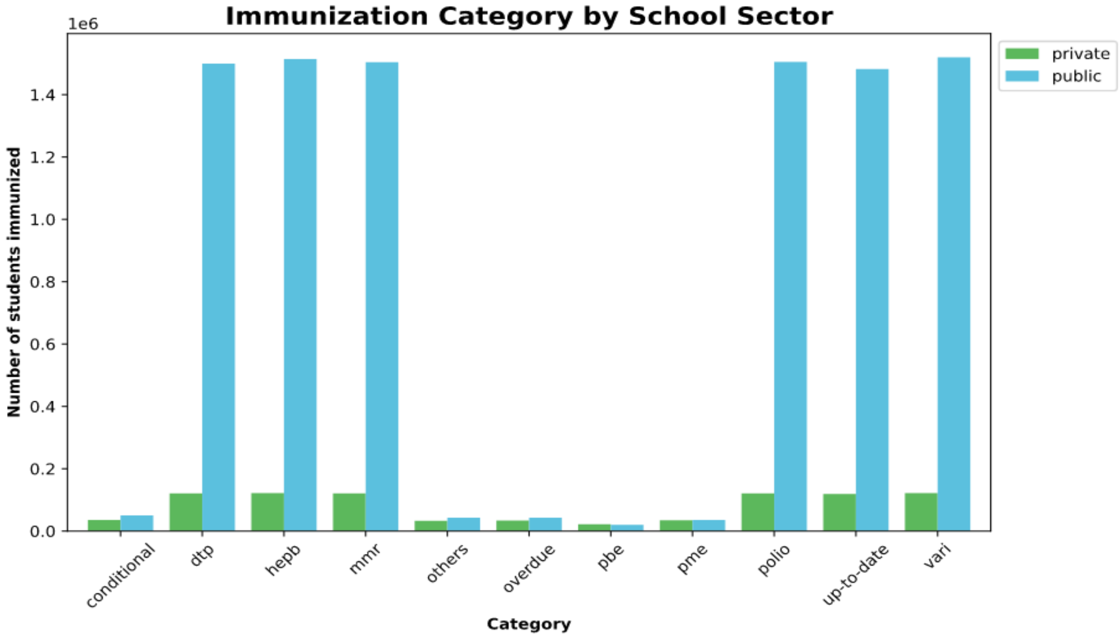


Figure 20 Overall immunizations of each type done per school category

The Figure 20 bar graph represents the number of students immunized by each category of the vaccine given the category of the school. It is evident that polio, measles, mumps and rubella (MMR), chickenpox (Varicella), diphtheria, tetanus and whooping cough (DPT) and hepatitis B (HEPB) are the most common diseases for which most students have been vaccinated. Others such as PBE and PME are less common so that gives an idea of which vaccinations are most needed and their required quantity.