In [1]:

```
# Import dependencies
import pandas as pd

# Ignore Python warnings about past model versions
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
# Import dataset and display first rows of dataset
df = pd.read_csv("data/middle_school_features.csv")
display(df)
```

	dbn	school_name	applications	acceptances	per_pupil_spending	avg_class_size	asia
0	01M034	P.S. 034 FRANKLIN D. ROOSEVELT	6	0	24890.0	20.15	
1	01M140	P.S. 140 NATHAN STRAUS	6	0	23536.0	24.56	
2	01M184	P.S. 184M SHUANG WEN	67	23	16206.0	29.69	
3	01M188	P.S. 188 THE ISLAND SCHOOL	0	0	21960.0	24.09	
4	01M301	TECHNOLOGY, ARTS, AND SCIENCES STUDIO	11	0	25444.0	15.80	
...	...	...	...	...	...	...	
589	84X538	ICAHN CHARTER SCHOOL 5	20	1	NaN	NaN	
590	84X703	BRONX PREPARATORY CHARTER SCHOOL	22	1	NaN	NaN	
591	84X704	KIPP ACADEMY CHARTER SCHOOL	23	1	NaN	NaN	
592	84X706	HARRIET TUBMAN CHARTER SCHOOL	24	1	NaN	NaN	
593	84X717	ICAHN CHARTER SCHOOL	24	1	NaN	NaN	

594 rows × 24 columns

Data Exploration and Cleaning

In [3]:

```
# Drop school ID code which is an unnecessary feature
df = df.drop("dbn", axis = 1)

# Display columns that have missing values and display sum of missing values
missing = df.isnull().sum()
missing[missing > 0]
```

Out[3]:

```
per_pupil_spending                 121
avg_class_size                     121
asian_percent                        2
black_percent                        2
hispanic_percent                     2
multiple_percent                     2
white_percent                        2
rigorous_instruction                43
collaborative_teachers              43
supportive_environment              43
effective_school_leadership         43
strong_family_community_ties        45
trust                               45
disability_percent                   2
poverty_percent                      2
ESL_percent                          2
school_size                          2
student_achievement                 47
reading_scores_exceed                8
math_scores_exceed                   8
dtype: int64
```

In this dataset, data for **per_pupil_spending** and for **avg_class_size** is missing for all charter schools. However given that the 109 charter schools represented in this dataset are all in the last rows of the dataset, I was able to split the dataset into two groups: public schools and charter schools. I then calculated the median value for public schools and then extrapolate this median value for all charter schools. Additionally there was missing data for those two rows for 12 public schools, which I also inputted the median values for both features for those schools with missing values.

In [4]:

```
# Split the dataset into school category based on row number
public_schools = df[:-109]
charter_schools = df[-109:]

# Compute median values for both missing categories for public schools
med_pupil_spending_public = public_schools["per_pupil_spending"].median()
med_avg_class_size_public = public_schools["avg_class_size"].median()

# Impute missing values for charter schools using median public school values
charter_schools["per_pupil_spending"].fillna(med_pupil_spending_public, inplace
charter_schools["avg_class_size"].fillna(med_avg_class_size_public, inplace = Tr
```

In [5]:

```
# Combine the public and charter school data back together
cleaned_df = pd.concat([public_schools, charter_schools])
```

```
# Determine how many missing values there are now for these two features
cleaned_df.isnull().sum()[["per_pupil_spending", "avg_class_size"]]
```

Out[5]:

```
per_pupil_spending    12
avg_class_size        12
dtype: int64
```

In [6]:

```
# Impute median values for public schools with missing values
cleaned_df["per_pupil_spending"].fillna(med_pupil_spending_public, inplace = Tru
cleaned_df["avg_class_size"].fillna(med_avg_class_size_public, inplace = True)

# Once again determine if there are any more missing values for these two featur
cleaned_df.isnull().sum()[["per_pupil_spending", "avg_class_size"]]
```

Out[6]:

```
per_pupil_spending    0
avg_class_size        0
dtype: int64
```

In [7]:

```
# Display other features which contain missing values
missing_features = missing[missing > 0].index

# Impute missing values within these columns using median value
for feature in missing_features:
    median = cleaned_df[feature].median()
    cleaned_df[feature].fillna(median, inplace = True)
```

In [8]:

```
# Check to see if there are any more missing values within the dataset
missing = cleaned_df.isnull().sum()
missing
```

Out[8]:

```
school_name                     0
applications                    0
acceptances                     0
per_pupil_spending              0
avg_class_size                  0
asian_percent                   0
black_percent                   0
hispanic_percent                0
multiple_percent                0
white_percent                   0
rigorous_instruction            0
collaborative_teachers          0
supportive_environment          0
effective_school_leadership     0
strong_family_community_ties    0
trust                           0
disability_percent              0
poverty_percent                 0
ESL_percent                     0
school_size                     0
student_achievement             0
reading_scores_exceed           0
math_scores_exceed              0
dtype: int64
```

In [9]:

```
# Export the cleaned dataset to be used for model training
path = "data/cleaned_school_data.csv"
cleaned_df.to_csv(path, index = False)
```