Freshman on Track: Fact or Fiction?

Team AutoBirdBrain
Paul Brown, Jess Greco, Michael O'Leary, John Leigh, Connor Thomas

Project Overview

Full write-up

OVERVIEW:

Using publicly available data form Chicago Public Schools, we set out to analyze the claim from UIC and the To&Through program that how a student performs during their freshman year of high school is the most accurate predictor of whether or not they will graduate.

Using predictive modeling, clustering, and PCA analysis, we discovered that this claim is, by and large, true.

However, issues with data availability, aggregation, and consistency prevented us from exploring our initial question: can we predict if a student will be “On Track” in their freshman year based on 8th grade testing scores and attendance data?

Data Gathering and Normalization

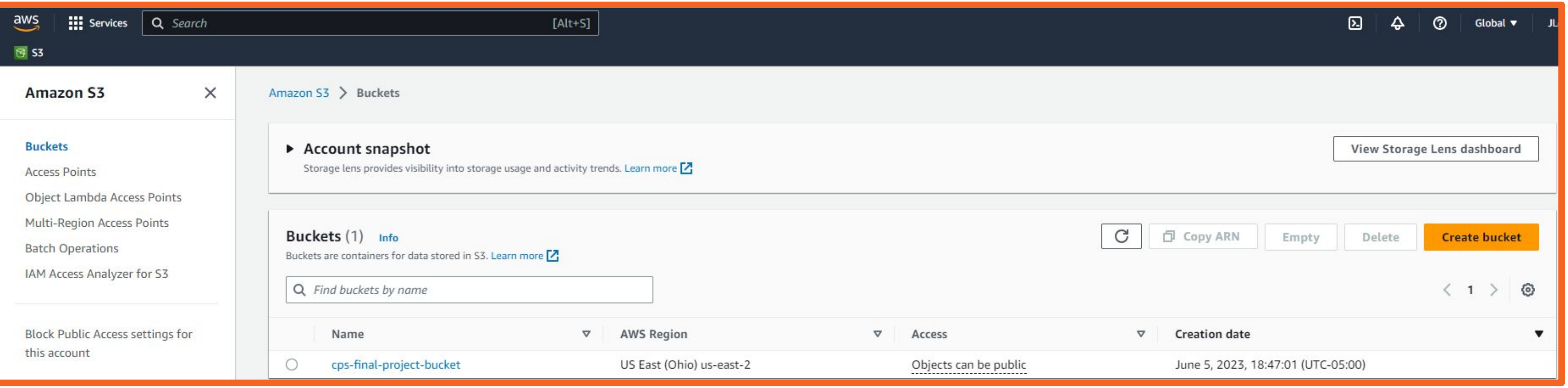
We gathered data from various sources including **Illinois State Board of Education** and the **Chicago Public Schools Data Portal.** Features for evaluation included:

- Percent Freshmen on Track,
- Expenditure per Student,
- School Type,
- Student Count,
- Demographic Percentages,
- Graduation Percent,
- Chronic Truancy Percent,
- Reading and Math RIT Scores

Sources:
https://www.isbe.net/Pages/Illinois-State-Report-Card-Data.aspx,
https://www.cps.edu/about/district-data/metrics/

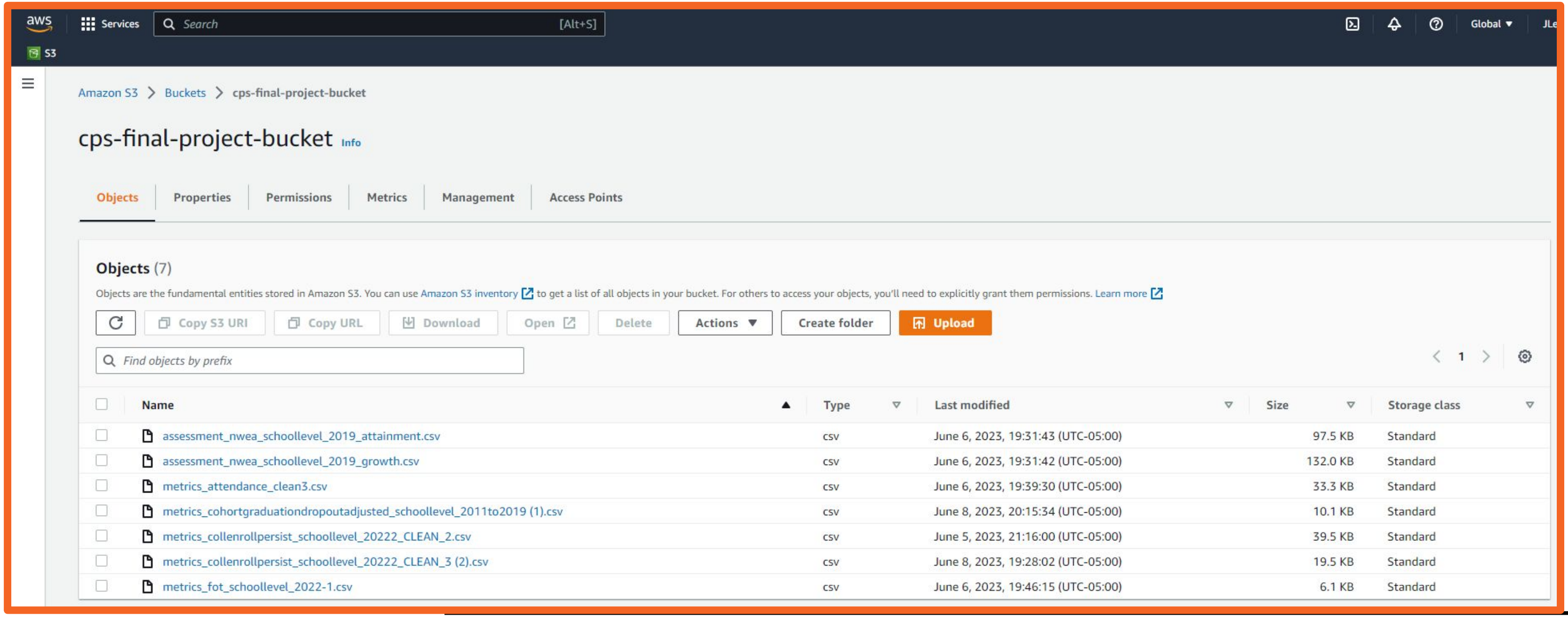
AWS S3 Bucket

- We created an Amazon AWS S3 Bucket to store our data



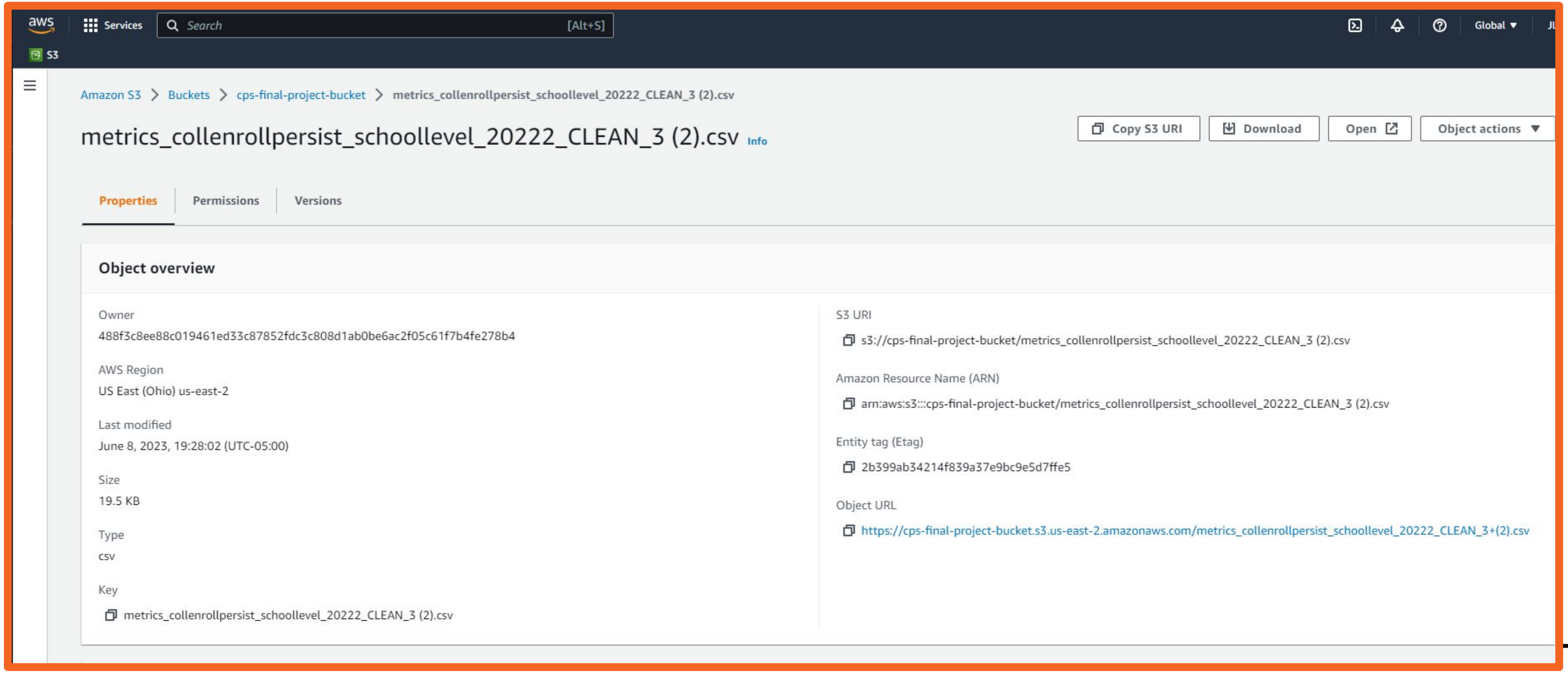
S3 data storage

- We stored pre-cleaned CSV's in our S3 bucket.



S3 Bucket - URL reference

- The CSV's were accessed through Spark by referencing the public facing URL that AWS generated for us (Object URL)



Google Colab Importation

```
[ ] from pyspark import SparkContext, SparkConf
    #Start Spark session
    from pyspark.sql import SparkSession
    spark = SparkSession.builder.appName("Final Project Analysis").getOrCreate()
```

```
from pyspark import SparkFiles
# Read in data from S3 Buckets
url20 = "https://cps-final-project-bucket.s3.us-east-2.amazonaws.com/metrics_collenrollpersist_schoollevel_20222_CLEAN_3+(2).csv"
spark.sparkContext.addFile(url20)
persist_3_df = spark.read.csv(SparkFiles.get("metrics_collenrollpersist_schoollevel_20222_CLEAN_3+(2).csv"), sep=",", header=True)

persist_3_df.show()
```

```
[ ] persist_3_df = persist_3_df.toPandas()
    persist_3_df.head()
```

	School_ID	Annualized School Name	status	Class of 2019 Graduates	Class of 2019 Enrollments	Class of 2019 Enrollment Pct	Class of 2019 # of Enrollments Persisting	Class of 2019 Persistence Pct	Class of 2018 Graduates	Class of 2018 Enrollments	...
0	400013	ASPIRA - EARLY COLLEGE HS	None	75	30	40	18	60	83	49	...
1	400022	CHIARTS HS	None	133	114	85.7	97	85.1	146	117	...
2	400032	CICS - ELLISON HS	None	101	62	61.4	31	50	72	47	...
3	400033	CICS - LONGWOOD	None	87	44	50.6	30	68.2	107	52	...
4	400034	CICS - NORTHTOWN HS	None	207	151	72.9	114	75.5	178	145	...

5 rows × 28 columns

Merging DataFrames

- In order to merge dataframes successfully, we spent a lot of time early on cleaning up CSVs by filtering out unwanted data, renaming columns, converting types, etc...
- Our final merged dataframe included the feature and target data for our data models to analyze, which we defined by assigning X and y variables to the relevant columns

```
[ ]  persist_merge_fot_df = pd.merge(persist_3_df, fot_df, on=['School ID'])
     persist_merge_fot_df.head()
```

School ID	Annualized School Name	status	Class of 2019 Graduates	Class of 2019 Enrollments	Class of 2019 Enrollment Pct	Class of 2019 # of Enrollments Persisting	Class of 2019 Persistence Pct	Class of 2018 Graduates	Class of 2018 Enrollments	Class of 2018 Enrollment Pct	...	SY 2019 On-Track Rate	SY 2019 Total Number of Freshmen	SY 2018 On-Track Rate	SY 2018 Total Number of Freshmen	SY 2017 On-Track Rate	SY 2017 Total Number of Freshmen	SY 2016 On-Track Rate	SY 2016 Total Number of Freshmen	SY 2015 On-Track Rate	SY 2015 Total Number of Freshmen
609674	CHICAGO VOCATIONAL HS	None	200	97	48.5	42	43.3	210	80	38.1	...	91.5	260	88.8	206	89.7	174	92.0	224	85.2	243
609676	DUNBAR HS	None	76	53	69.7	29	54.7	99	57	57.6	...	90.3	62	81.4	86	78.4	102	75.5	147	62.4	186
609678	JONES HS	None	435	383	88	352	91.9	463	400	86.4	...	98.0	509	99.6	452	99.1	423	99.1	428	99.0	482
609679	PROSSER HS	None	298	211	70.8	133	63	293	224	76.5	...	82.5	275	92.4	397	95.8	355	94.7	356	86.6	357
609680	PAYTON HS	None	213	198	93	184	92.9	219	202	92.2	...	99.7	310	98.8	336	99.3	294	99.6	229	96.9	229

5 rows × 38 columns

Feature Importance

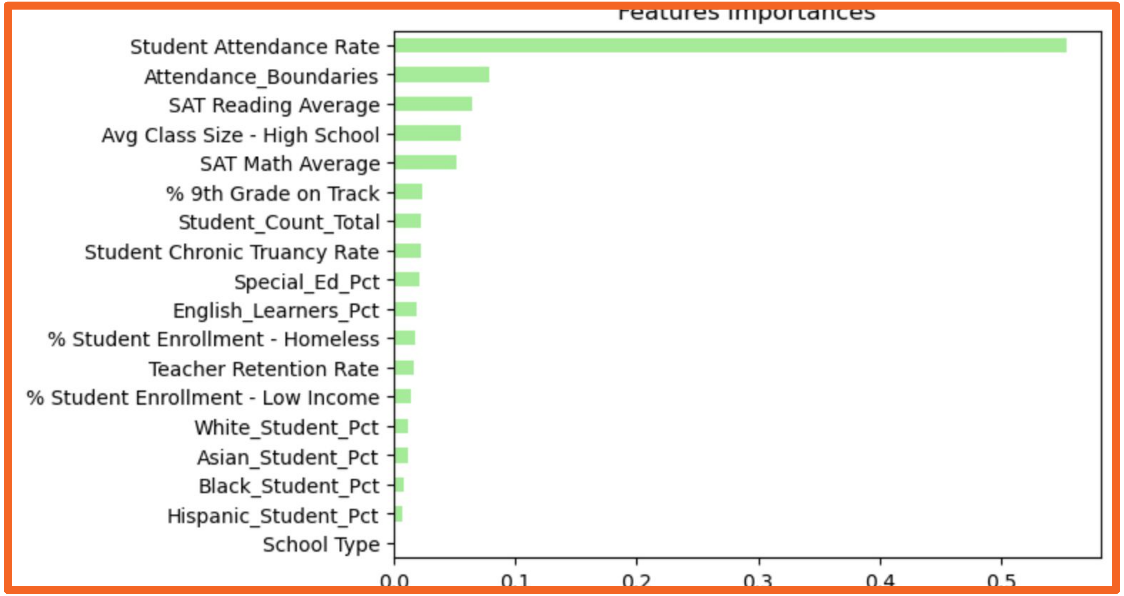
```
[ ] # Split the data into training and testing sets
    X = merge_df.drop('High School Graduation Rate', axis=1)
    y = merge_df['High School Graduation Rate']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
```

```
[ ] # Scale the numerical variables
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
    # Train the model
    model = RandomForestRegressor(n_estimators=100, random_state=1)
    model.fit(X_train, y_train)
```

```
[ ] # Get feature importances
    importances = model.feature_importances_
    feature_importances = pd.Series(importances, index=X.columns).sort_values(ascending=False)
    print(feature_importances)
```

We attempted to build a Random Forest model to find the most important features contributing to High School Graduation Rate

Feature Importance



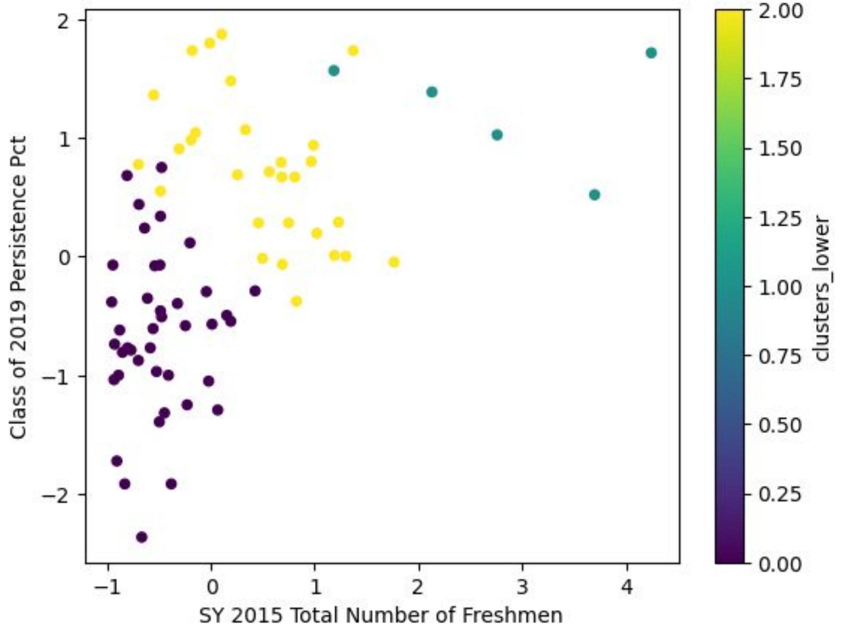
What we found was that the Student Attendance Rate was the most important feature by far in contributing to the High School Graduation Rate.

The model showed that factors such as a student's race and school type had little importance.

Clustering

```
persist_predictions.plot.scatter(x="SY 2015 Total Number of Freshmen",
                                 y="Class of 2019 Persistence Pct",
                                 c="clusters_lower",
                                 colormap='viridis')
```

```
<Axes: xlabel='SY 2015 Total Number of Freshmen', ylabel='Class of 2019 Persistence Pct'>
```



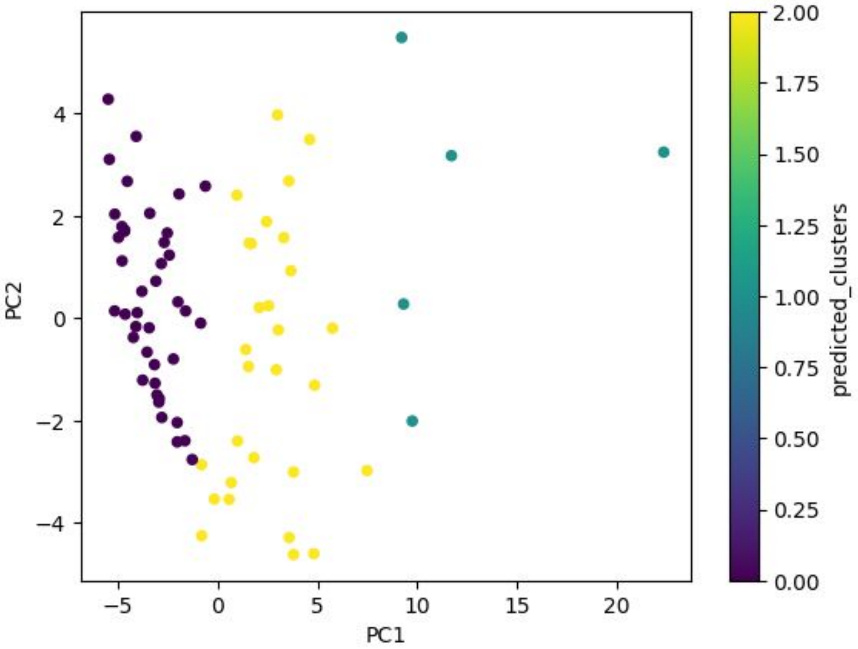
3 clusters:

- **Blue**: high enrollment and high persistence
- **Yellow**: middle enrollment and middle persistence
- **Purple**: low enrollment and low persistence

Some schools in the yellow cluster have higher persistence than the blue cluster.

```python
df_pca_predictions.plot.scatter(x="PC1",
                                y="PC2",
                                c="predicted_clusters",
                                colormap='viridis')
```

```
<Axes: xlabel='PC1', ylabel='PC2'>
```



Principal Component Analysis

With n=3 components, we obtain an explained_variance_ratio of 0.898

Principal Component Analysis

```
comps = []
for n in range(3):
  primaries = []
  for i in range(len(abs( pca.components_ )[n])):
    if abs( pca.components_ )[n][i] > 0.25:
      primaries.append(i)
  comps.append(primaries)
print('---------')
for comp in comps:
  for i in comp:
    print(persist_drop.columns[i])
  print('---------')
```

```
---------
---------
SY 2019 On-Track Rate
SY 2018 On-Track Rate
SY 2017 On-Track Rate
SY 2016 On-Track Rate
SY 2015 On-Track Rate
---------
SY 2019 On-Track Rate
SY 2018 On-Track Rate
SY 2017 On-Track Rate
SY 2016 On-Track Rate
SY 2015 On-Track Rate
---------
```

Setting a cutoff value of **0.25**, we can see that the main contributing components are the On-Track rates for **2015-2019**.

This matches the results from University of Chicago's inquiry

Logistic (turned linear) Regression

```python
# Import the LogisticRegression module from SKLearn
from sklearn.linear_model import LogisticRegression

# Instantiate the Logistic Regression model
# Assign a random_state parameter of 1 to the model
logistic_regression_model = LogisticRegression(solver='lbfgs', random_state=1)

# Fit the model using training data
logistic_regression_model.fit(X_train, y_train)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-72-41ce87eccaf8> in <cell line: 9>()
      7
      8 # Fit the model using training data
----> 9 logistic_regression_model.fit(X_train, y_train)

1 frames
/usr/local/lib/python3.10/dist-packages/sklearn/utils/multiclass.py in check_classification_targets(y)
    216             "multilabel-sequences",
    217         ]:
--> 218         raise ValueError("Unknown label type: %r" % y_type)
    219
    220

ValueError: Unknown label type: 'continuous'
```

SEARCH STACK OVERFLOW

Model Set Up

TARGET VALUES:

X: Graduation rate (%)
Y: FOT Rate (%)

```
grad_rate_merge_fot_df['2016 Grad. %'].astype(float)
```

```
# Convert column of interest to float
grad_rate_merge_fot_df['SY 2016 On-Track Rate'].astype(float)
```

```
# Separate the data into labels and features

# Separate the y variable, the labels
y = grad_rate_merge_fot_df['SY 2016 On-Track Rate']

# Separate the X variable, the features
X = grad_rate_merge_fot_df.copy()
X = X[['2016 Grad. %']]
```

Model Accuracy

Mean Absolute Error (MAE): 8.281

Mean Squared Error (MSE): 103.251

Root Mean Squared Error (RMSE): 10.161

R-squared: 0.292 Adjusted

R-squared: 0.255

Cross validation scores: [0.021, 0.133, -1.101, 0.338, 0.118]

Mean cross validation score: -0.098

```python
# Make a prediction using the testing data
test_predictions = linear_regression_model.predict(X_test)
comparison_df = pd.DataFrame({'Predictions': test_predictions, 'Actual': y_test})
print(comparison_df)
```

	Predictions	Actual
11	85.283022	92.8
75	84.702993	78.0
65	84.771232	71.4
38	86.340722	81.5
91	79.260957	57.9
85	88.763196	95.0
39	85.231843	90.5
43	86.971930	94.2
53	86.118946	83.0
62	88.149047	94.9
86	87.432541	83.1
76	83.457637	87.1
67	83.952367	71.4
74	88.097868	87.8
59	79.073300	53.3
71	87.466660	96.7
36	88.609659	95.0
41	84.447098	75.3
32	78.697988	83.3
48	87.756675	99.3
45	88.097868	84.1

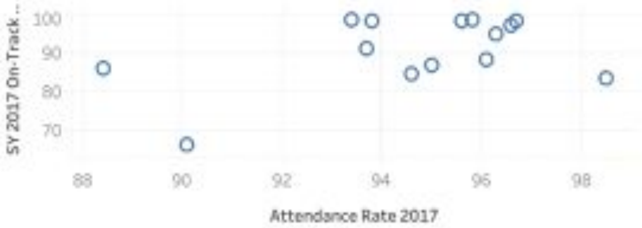
The Relationship Between Attendance and Performance in Chicago Public Schools

The purpose of this study was to use publicly available data to answer a simple question, does student attendance affect student performance in Chicago Public Schools in any quantifiable way. We believe that our findings provided an answer to this question.

2015 Attendance/OT

2016 Attendance/OT

2017 Attendance/OT

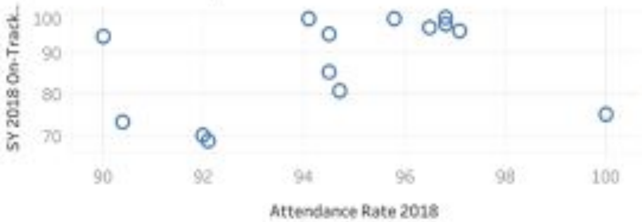


Linear Regression Analysis

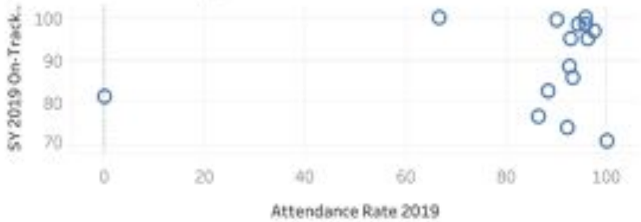
Using linear regression to provide a visual summary of our findings- we found that there is a linear relationship between the overall rate of attendance and student performance, measured by the Freshman On-Track score (FOT). As expected, the FOT rose along with the Attendance Rate- up to a point.

As demonstrated in the chart below, there were several outliers and the correlation between the dependent variable (FOT) and the independent variable (Attendance) in this model appears to weaken past a certain percentage. So Attendance appears to be a driver for FOT up to a certain point. The reason for this change in correlation may be due to several known or unknown factors, but there are also concerns that inconsistency in the raw data may have had some effect on the findings as well.

2018 Attendance/OT



2019 Attendance/OT



Summary

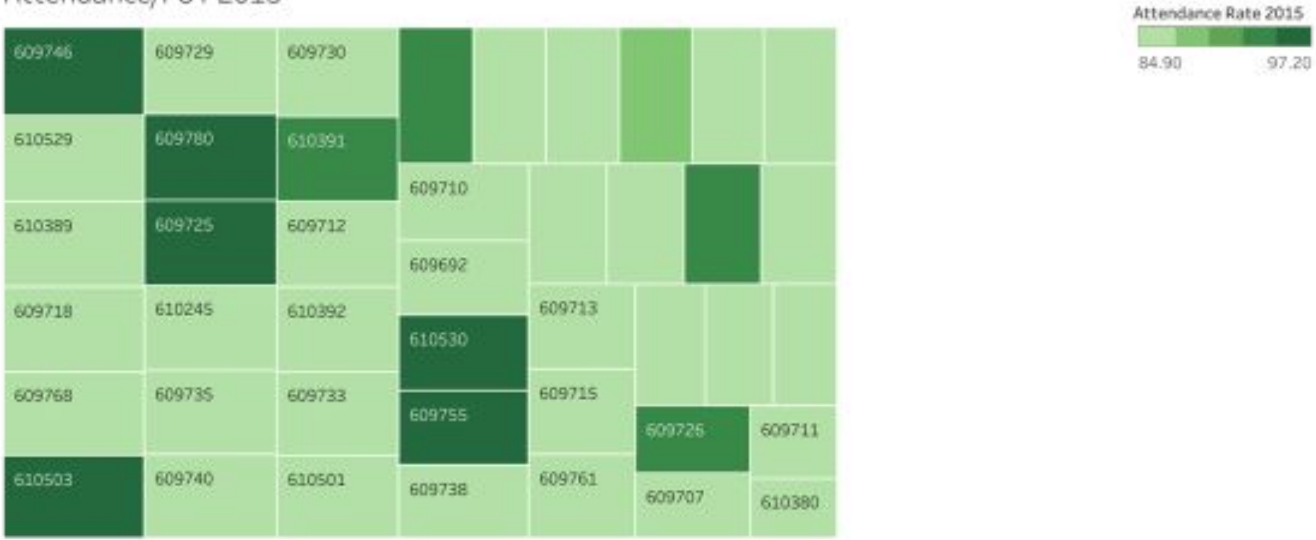
When the yearly data is compared side by side, the pattern appears to repeat during the years being observed (2015-2019), as shown in the previous charts. This is an unexpected and curious phenomenon, and may deserve a deeper exploration than our alloted time permitted.

The overall results however, appear to support our initial hypothesis- school attendance has a positive correlation to student performance. As the attendance rate increases, so to does the performance, measured by the Freshman On-Track score. The relationship appears to be stronger and more pronounced where the attendance rate nears 90%. Above the 90% range, we begin to see more outliers and exceptions and the corelation appears to weaken.

In addition, it may be likely that linear regression might not be the best analytical tool for analyizing this particular collection of data. After a certain point, a non-linear technique may be more appropriate for determining the relationship.

This project drew from publicly available data provided by Chicago Public Schools and the State of Illinois. While there was an abundance of public data about overall test scores and attendance, we found that some of the information was obscured or was intentially left opaque to ensure student anonymity, and in some cases to provide some form of cover for lower performing schools. We also found that school closures and openings throughout the observation period, along with several exceptions to grade level and program size, very likely skewed the results and did not offer many viable options for detection and correction for these outliers.

Attendance/FOT 2015



Next Steps and Discussion

Issues with data

- Current data only shows information by school instead of by student
  - This is due to ethics concerns and to avoid tracking specific students (i.e. Male student of Hispanic descent transferred from (low-pop.) school A to (low-pop.) school B)
- ***How can we anonymize student data?***

Anonymizing student data

- Publish data from large population schools where a reasonable degree of anonymity can be established within the student body
- Publish data after a certain period of time following graduation (ex. 5-10 years past)

Next Steps and Discussion

Having individualized data will allow us to track progress by student instead of by school. With this data, we could potentially find predictors for whether a student will be in track or not by their freshman year. According to UofC:

"On-track students are more than three and one-half times more likely to graduate from high school in four years than off-track students"

Finding these students before their freshman year has the potential to set them on track before significant progress has been lost.

Questions?