Clustering Transit-Rail Data

David Booker-Earley

Presentation Outline

- Intro
  - Problem Statement and Goal
  - Data Overview
- Exploration
  - “Target” or Feature of Interest
- Clusters
  - Unsupervised ML Algorithms
  - Silhouette Scores
  - Cluster Visualization
- Next Steps
- Appendix

The Problem

- Essential Workers
  - Commute via train amid pandemic
- Social Distancing (SD)
  - Limits passengers per train
- Train Times
  - Need to be accurate
  - Need to account for SD

The Problem

- Essential Workers
  - Commute via train amid pandemic
- Social Distancing (SD)
  - Limits passengers per train
- Train Times
  - Need to be accurate
  - Need to account for SD

"What could possibly go wrong?"

- Delayed Trains!
  - Occurred before pandemic & SD
  - Will likely occur during
- May cause more problems
  - Active workers stuck at stations
  - Productivity hindered (again)
- Need to account for delays amid pandemic!

The Problem

*.. Safety first, but where is **my** train?!*

.. you said *"Arriving in 2 Minutes"* ***15 minutes ago!***

Approaching the Problem

1. *Which days of the week are typically the busiest?*
   .. and for which bus-route or rail-line?

2. *Which days typically have the most frequent and longest delays?*

3. *How can data from previous years be used to project the number of "in-service" vehicles needed during the pandemic?*

Goal

- ➢ Discover how rail-service-data can be grouped.
  - ○ No real “target” variable → Focus on delay time per day

How?

- ➢ Clustering!
  - ○ Thank you, unsupervised machine learning algorithms!

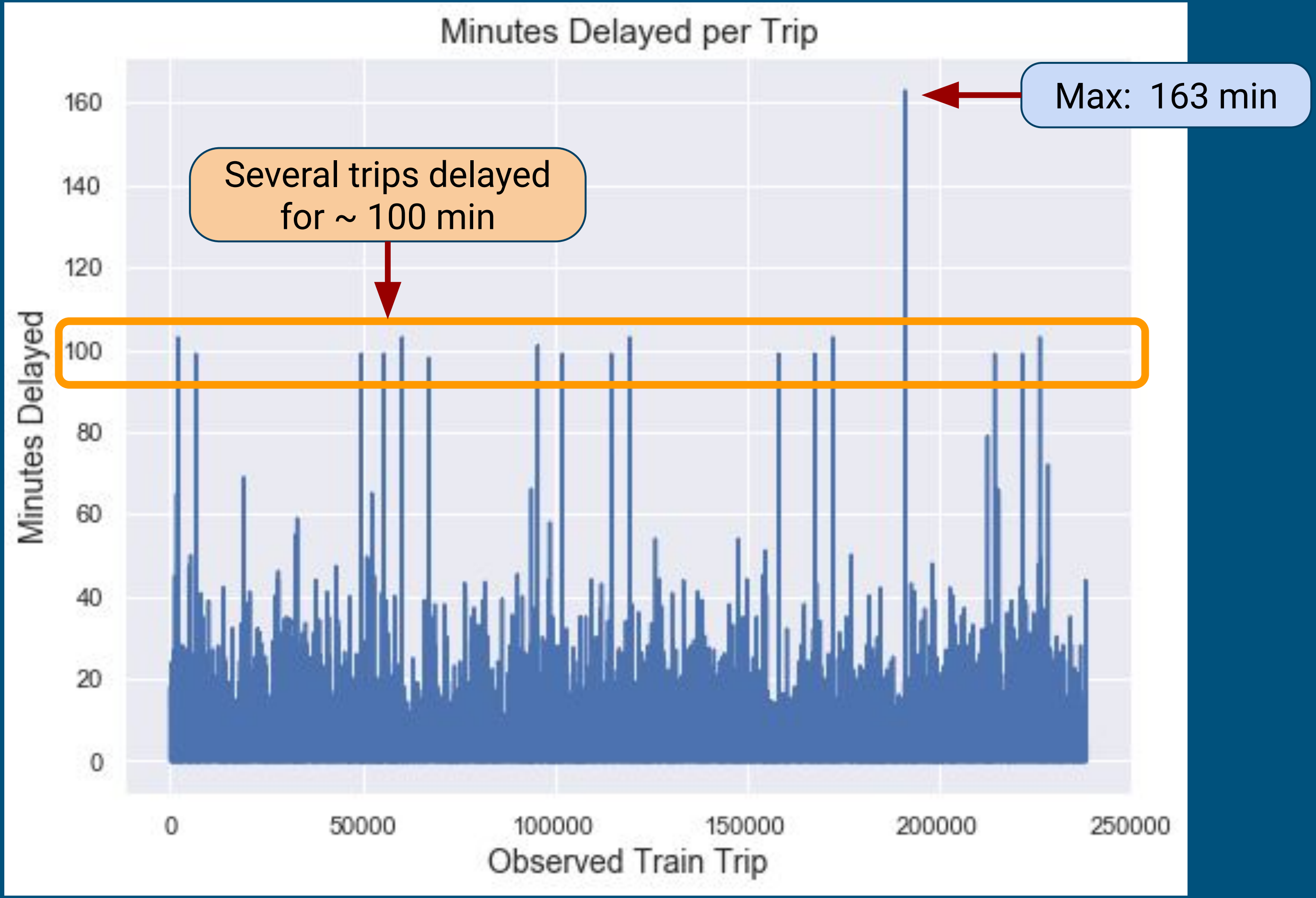
Data Overview

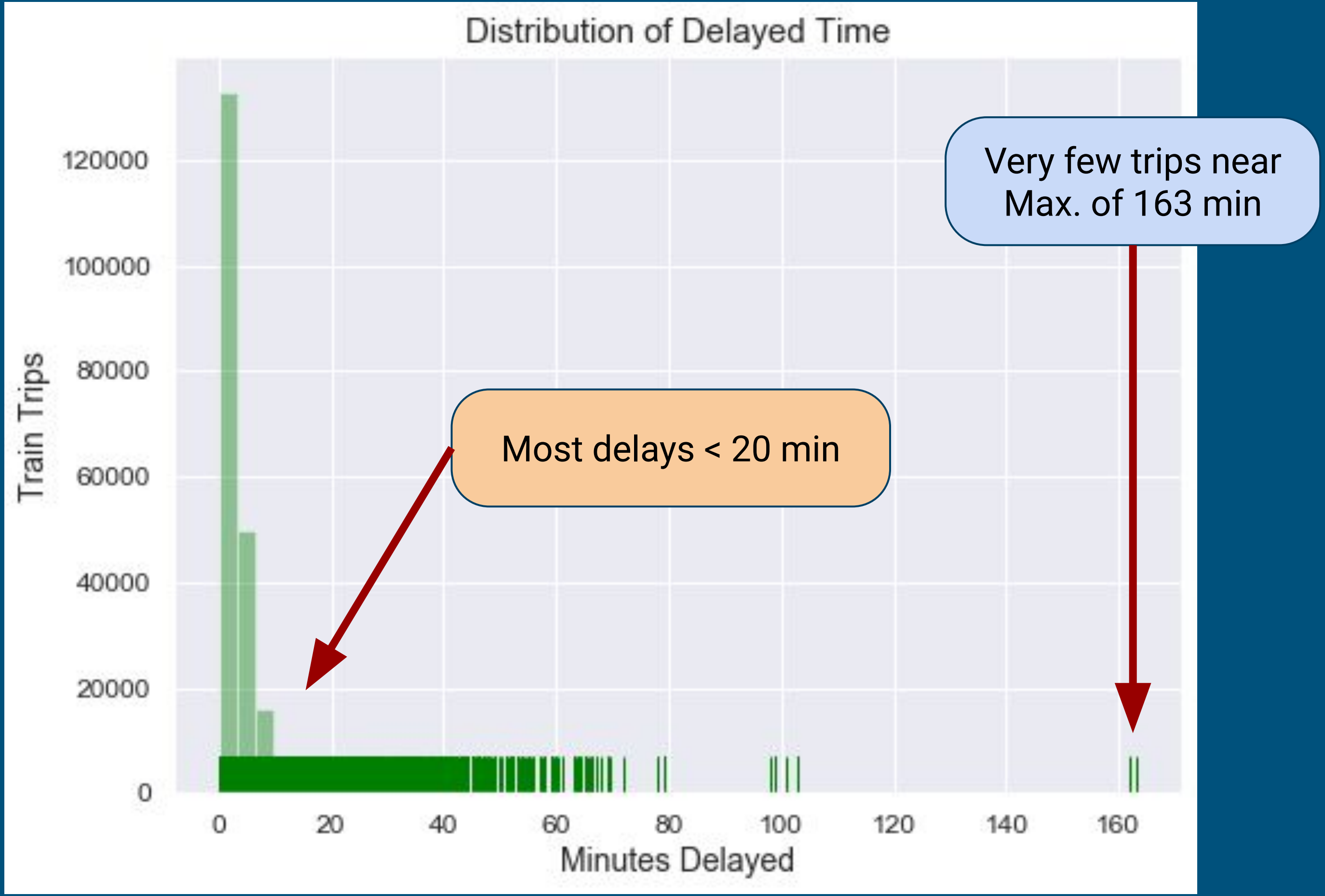
NJ Transit and Amtrak (NEC) Rail Performance Data

- ➢ Located on Kaggle
- ➢ Provides trip-level performance data for various months
  - ○ Date, Train ID, Destination, Delayed Time, etc.
- ➢ Selected Data → April, 2019
  - ○ 238,693 trip-entries
  - ○ 13 columns

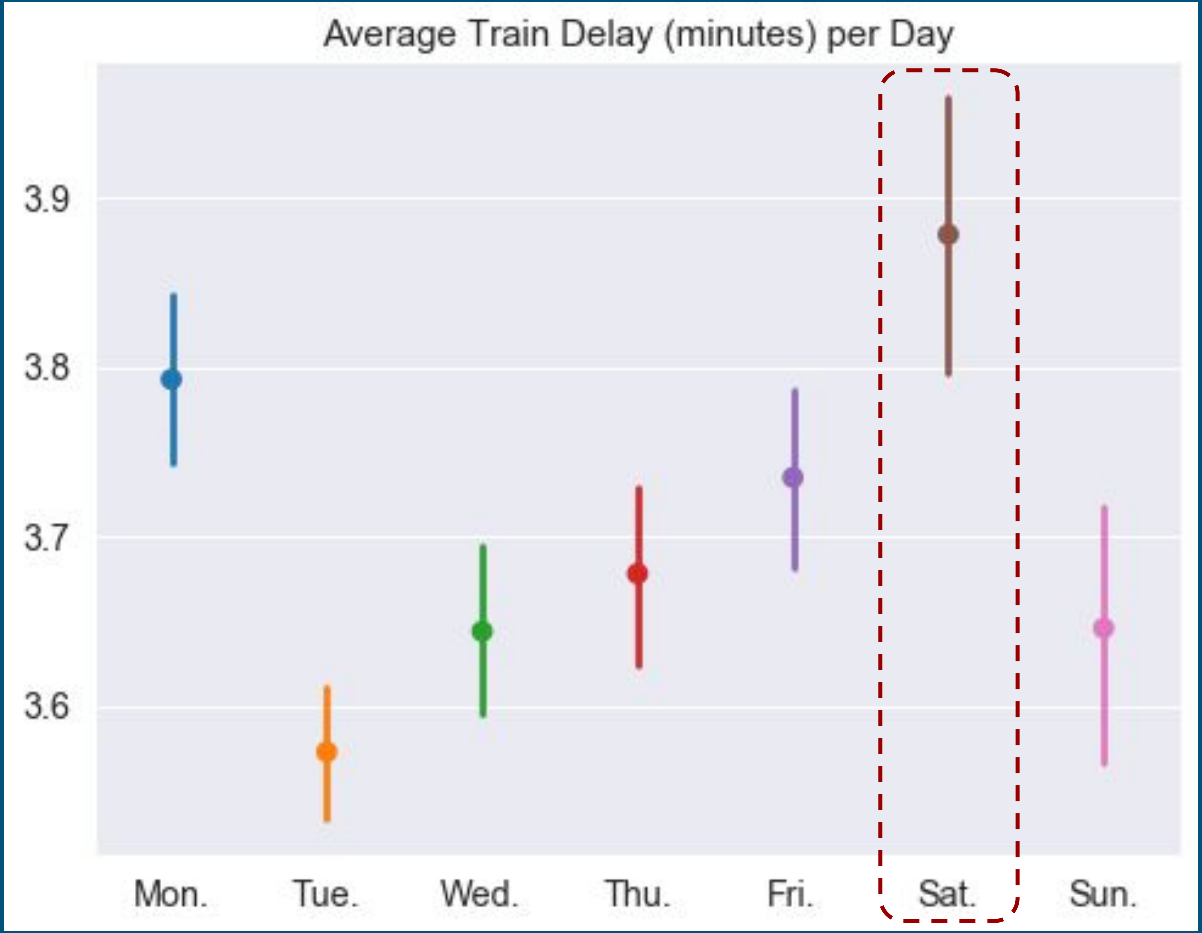
Exploration!

.. How late were the trains?


Minutes Delayed per Trip
Max: 163 min
Several trips delayed for ~ 100 min
Minutes Delayed
160
140
120
100
80
60
40
20
0
0
50000
100000
150000
200000
250000
Observed Train Trip


Distribution of Delayed Time
Train Trips
120000
100000
80000
60000
40000
20000
0
0
20
40
60
80
100
120
140
160
Minutes Delayed
Very few trips near Max. of 163 min
Most delays < 20 min

On average, which day of the week had the longest delay?



Average Train Delay (minutes) per Day
3.9
3.8
3.7
3.6
Mon.
Tue.
Wed.
Thu.
Fri.
Sat.
Sun.

Clustering!

Let's jump right in!

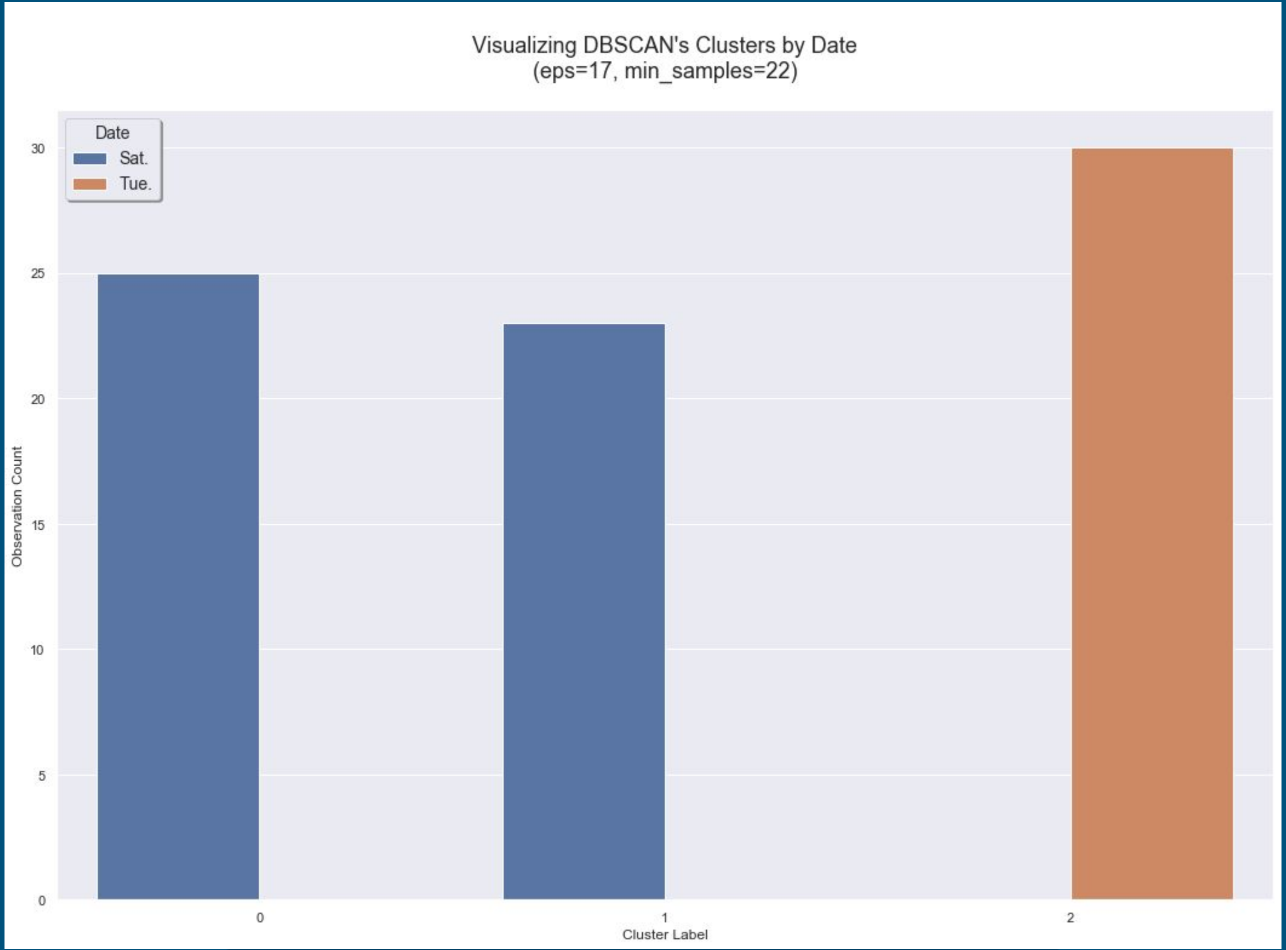
How well did the algorithms perform?

Best Variations of each Clustering Algorithm

```
Clustering with DBSCAN, eps=17, min_samples=22
----------------------------------------------
Estimated number of clusters (excluding noise): 3
Number of samples marked as noise: 213085
Silhouette score: -0.6085516
```

```
Clustering with KMeans, k=3
Silhouette score: 0.5875917
```

How many trips are there in each cluster with respect to days of the week?


Visualizing DBSCAN's Clusters by Date
(eps=17, min_samples=22)
Date
Sat.
Tue.
Observation Count
30
25
20
15
10
5
0
0
1
2
Cluster Label


Visualizing K-Means's Clusters by Date
(k=3)
Observation Count
16000
14000
12000
10000
8000
6000
4000
2000
0
0
1
2
Cluster Label
Date
Mon.
Tue.
Wed.
Thu.
Fri.
Sat.
Sun.

.. Remember

On average,

- Saturday had the longest delays.
- Monday had the second longest delays.
- Tuesday had the shortest delays.
- The other four days had similar delay durations.

Visualizing K-Means's Clusters by Date
(k=3)
Observation Count
16000
14000
12000
10000
8000
6000
4000
2000
0
0
1
2
Cluster Label
Date
Mon.
Tue.
Wed.
Thu.
Fri.
Sat.
Sun.

As always → Further research is needed

- Ridership per train line or destination.
- Delays due to overcrowded trains under normal conditions.
- Delays due to service malfunctions or accidents.
- Delays due to weather patterns.
- etc.

Next Steps

Next Steps for Further Research

1. Explore various aspects of emergencies, public events, and weather data for the month of April, both in 2019 and 2020.
2. Investigate how NJ Transit rail-data have changed from April of 2019 through May of 2020 (to find more patterns amid the pandemic).
3. Use clustering algorithms with additional data (more customer records, more factors, etc.) to group by more underlying patterns.
4. Use supervised machine learning models to predict which trains will be delayed or canceled.
5. Expand on evaluation metrics and tools for all algorithms used.
6. Based on those research results, discuss the newly discovered patterns surrounding train delays.

Thank you for your time!

Any questions?

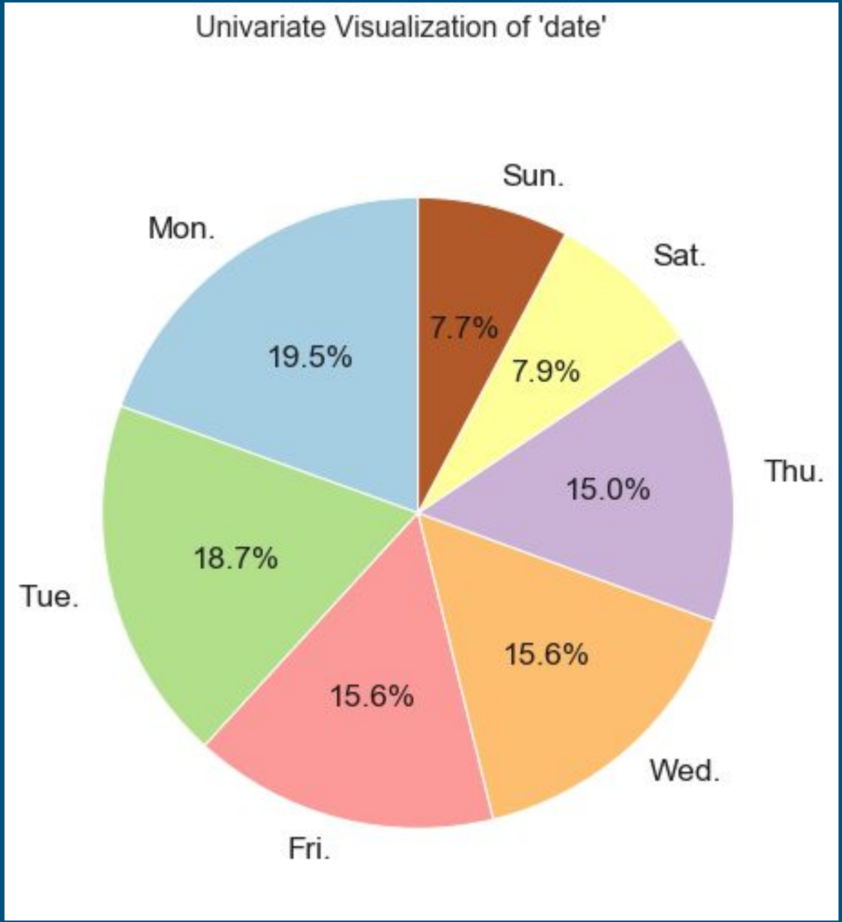
Appendix

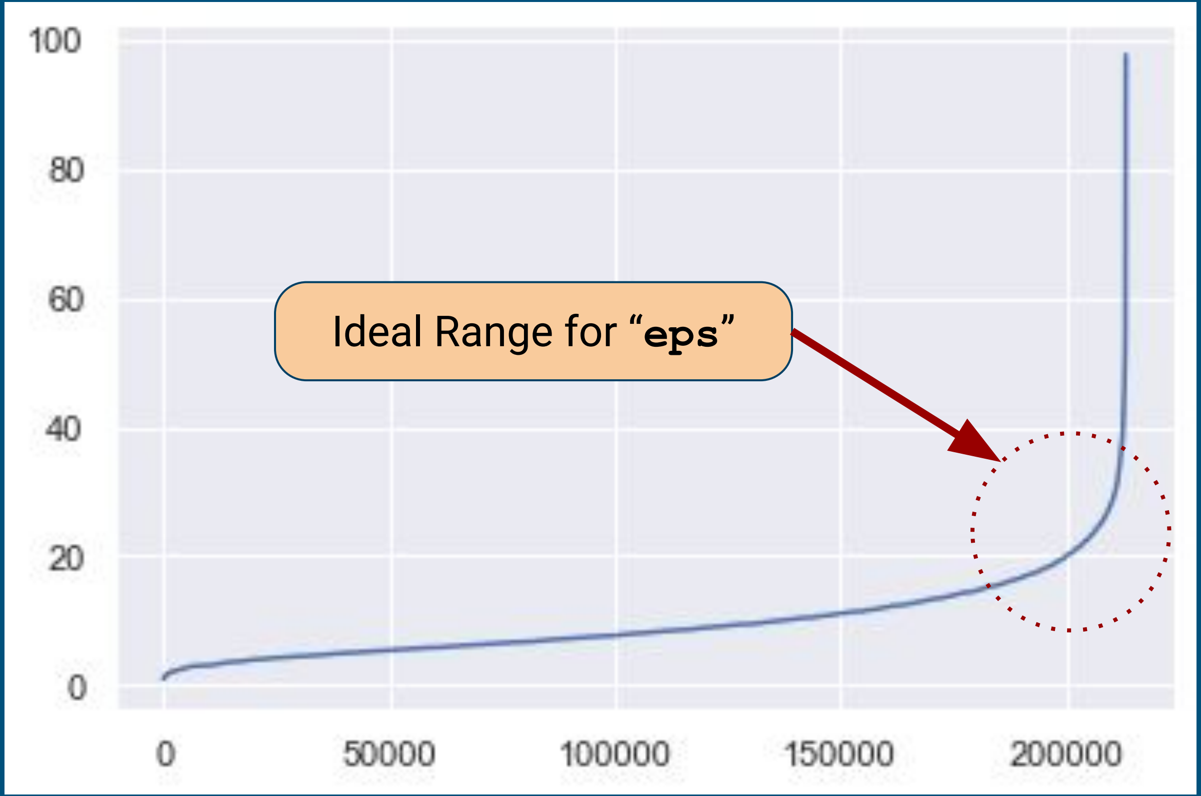
Who is this project for?

- Chief Executive Officers
- Data Scientists
- Machine Learning enthusiasts
- Transportation Providers
- .. anyone who is inherently inquisitive :)

Exploration

The dataset consists mostly of weekday trips.





Determining "`eps`" parameter for `DBSCAN` from elbow

DBSCAN → Determining the best "`min_sample`"

```
Clustering with DBSCAN, eps=17, min_samples=12
----------------------------------------------------
Estimated number of clusters (excluding noise): 172
Number of samples marked as noise: 210341
```

```
Silhouette score: -0.9590646972699022
```

DBSCAN → `eps=17, min_samples=12`

DBSCAN → Determining the best "min_sample"

```
Clustering with DBSCAN, eps=17, min_samples=20
------------------------------------------------------
Estimated number of clusters (excluding noise): 7
Number of samples marked as noise: 212979
Clusters (noise labeled as -1): [-1  0  1  2  3  4  5  6]
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Silhouette score: -0.8402470566445556
```

DBSCAN → `eps=17, min_samples=20`

DBSCAN → Determining the best "`min_sample`"

```
Clustering with DBSCAN, eps=17, min_samples=21
-----------------------------------------------
Estimated number of clusters (excluding noise): 5
Number of samples marked as noise: 213038
Clusters (noise labeled as -1): [-1  0  1  2  3  4]
* * * * * * * * * * * * * * * * * * * * * * * * * * *
Silhouette score: -0.81609333690897
```

DBSCAN → `eps=17, min_samples=21`

Determining the best DBSCAN variation

```
Clustering with DBSCAN, eps=20
------------------------------------------------------------
Estimated number of clusters (excluding noise): 8
Number of samples marked as noise: 212928
Clusters (noise labeled as -1): [-1  0  1  2  3  4  5  6  7]
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Silhouette score: -0.8526845709343379
```

DBSCAN → `eps=20, min_samples=22`

Determining the best DBSCAN variation

```
Clustering with DBSCAN, eps=22
-------------------------------------------------------------------------
Estimated number of clusters (excluding noise): 13
Number of samples marked as noise: 212756
Clusters (noise labeled as -1): [-1  0  1  2  3  4  5  6  7  8  9 10 11 12]
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Silhouette score: -0.9232809316514823
```

DBSCAN → `eps=22, min_samples=22`

Determining the best DBSCAN variation

```
Clustering with DBSCAN, eps=25
---------------------------------------------------------------------------
Estimated number of clusters (excluding noise): 17
Number of samples marked as noise: 212578
Clusters (noise labeled as -1): [-1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Silhouette score: -0.9296240524779781
```

DBSCAN → `eps=25, min_samples=22`

Clusters by DBSCAN → `eps=17, min_samples=22`

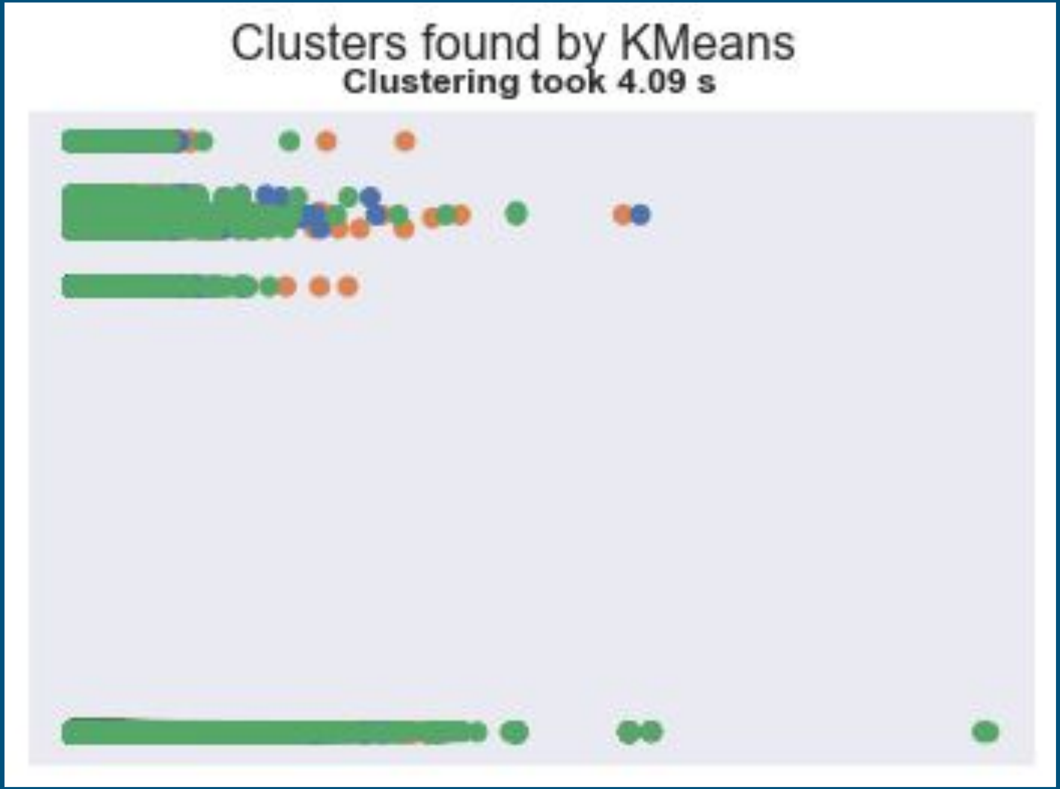
```
Analysis of K-Means
* Calculating the Relative Percent Difference (RPD) of Silhouette Scores

RPD for ('score_k3', 'score_k4'): 3.08%
RPD for ('score_k3', 'score_k7'): 7.01%
RPD for ('score_k3', 'score_k18'): 11.38%
RPD for ('score_k4', 'score_k7'): 3.93%
RPD for ('score_k4', 'score_k18'): 8.31%
RPD for ('score_k7', 'score_k18'): 4.38%
```

Silhouette Scores: K-Means

```
Clustering with KMeans, k=3
Silhouette score: 0.5875917364610415
```

Highest (best) score

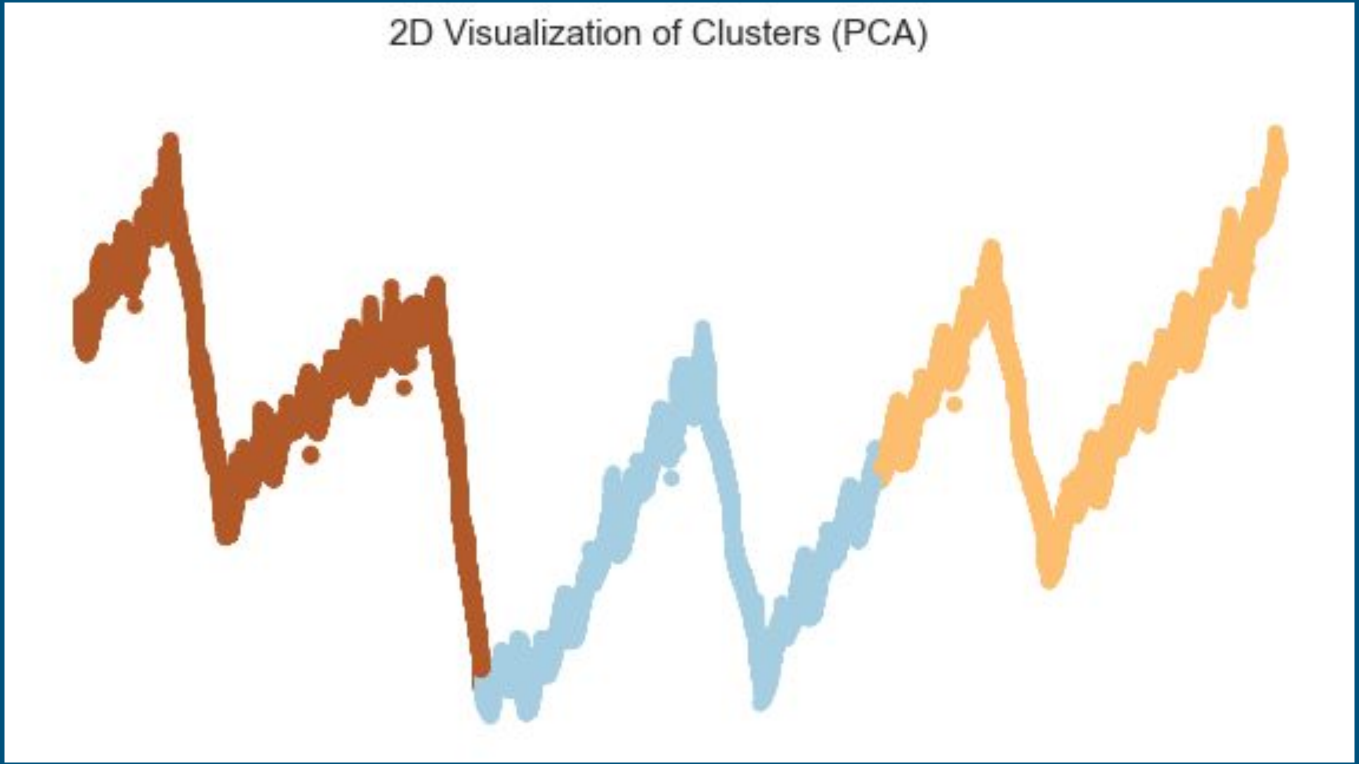
```
Clustering with KMeans, k=4
Silhouette score: 0.5697794562801087


Clustering with KMeans, k=7
Silhouette score: 0.5478048586935242


Clustering with KMeans, k=18
Silhouette score: 0.5243039735061048
```



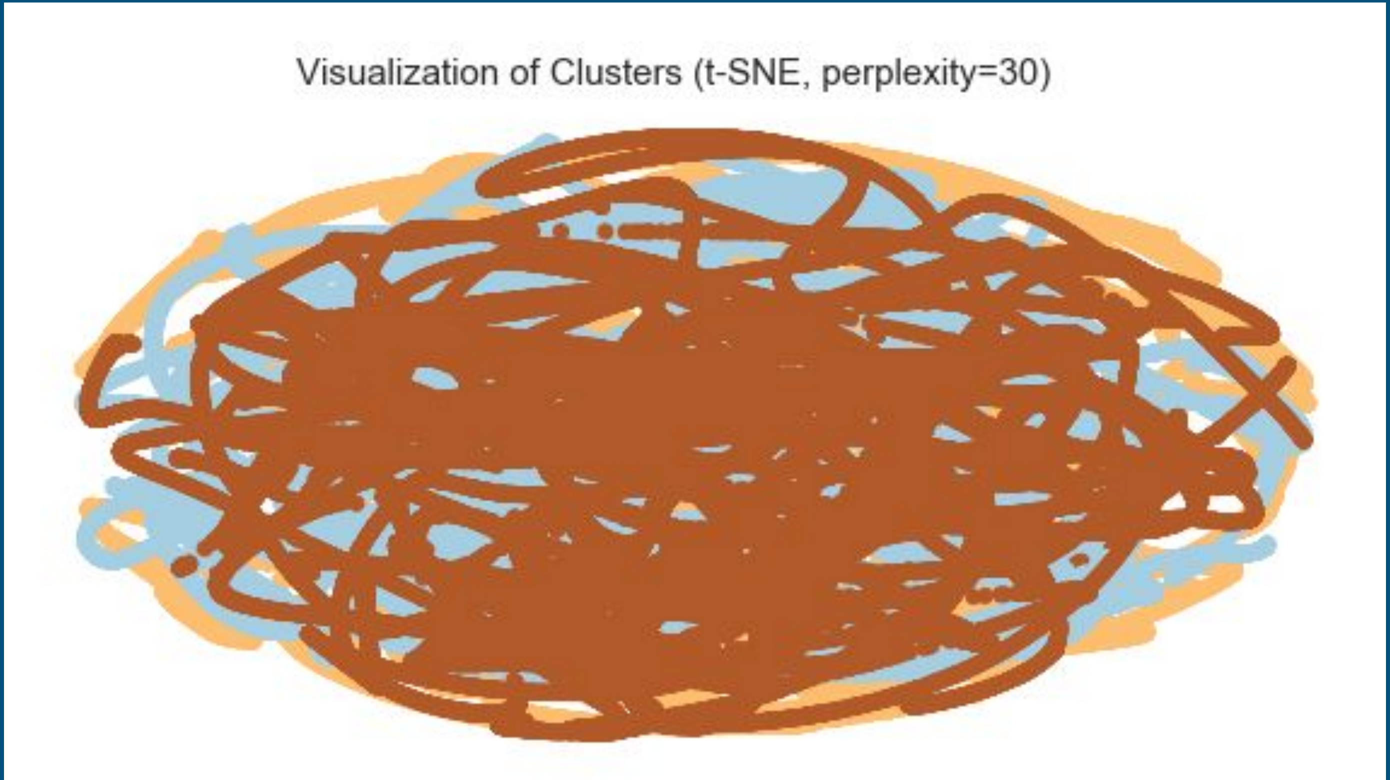
Clusters by K-Means → k=3

```
Confirm Updates from PCA (components = 2)
Old Shape: (213163, 10)
New Shape: (213163, 2)

-------------------------------------------------------------

Finding Collinearity among Features

The percentage % of "total variance in the dataset" captured
and explained by each principal component:

[9.99889509e+01 6.97200177e-03]
Est. Total: 100%
```

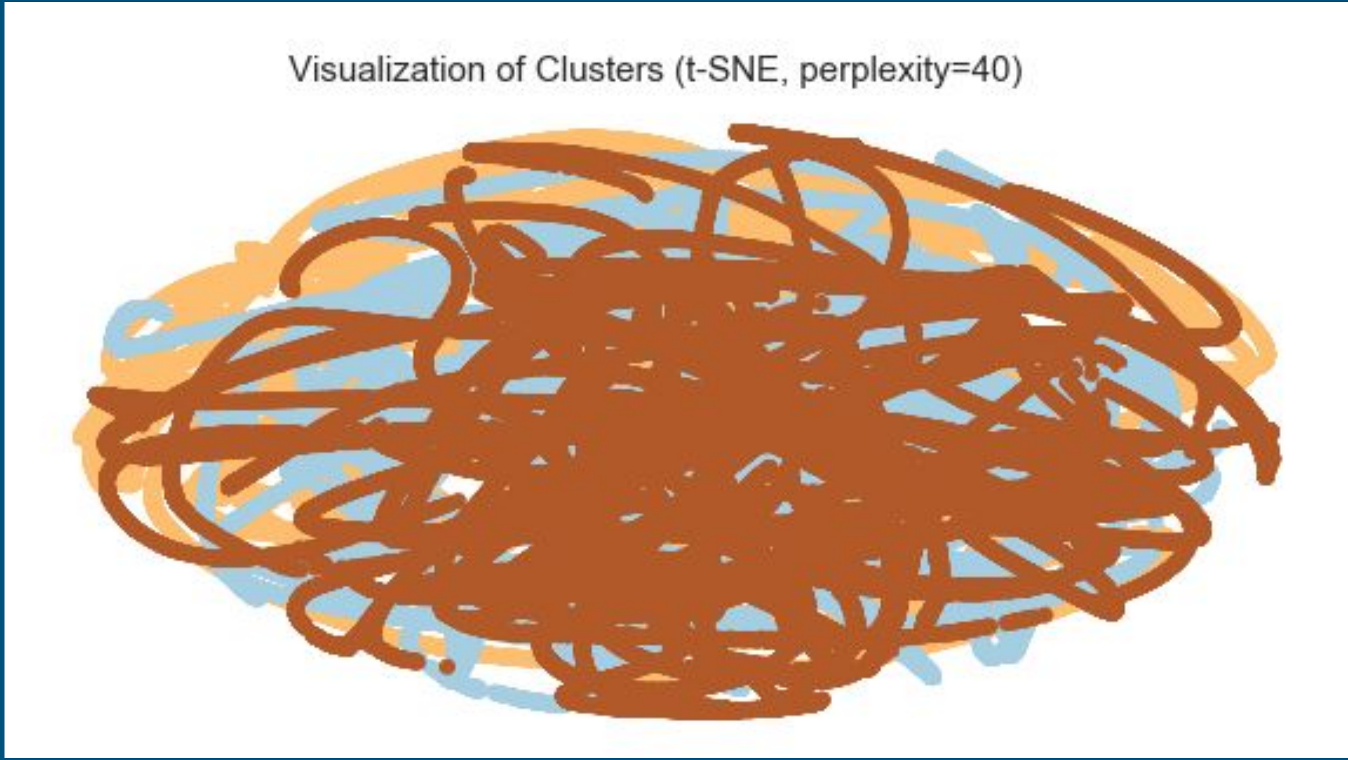
Cluster Visualization through Dimensionality Reduction → PCA



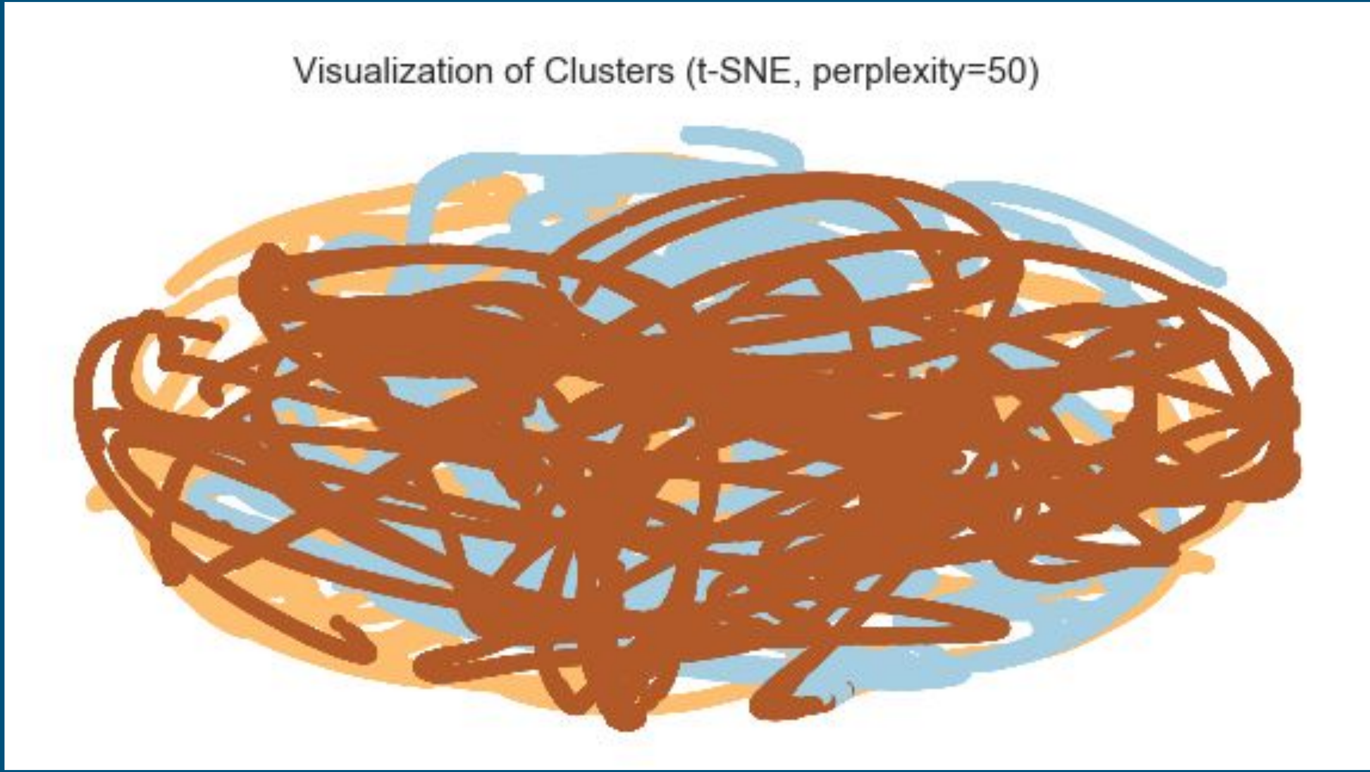
Cluster Visualization through Dimensionality Reduction → PCA



Cluster Visualization through Dimensionality Reduction → t-SNE (perplexity=30)



Cluster Visualization through Dimensionality Reduction → t-SNE (perplexity=40)



Cluster Visualization through Dimensionality Reduction → t-SNE (perplexity=50)

Using the Silhouette coefficient to evaluate the performance of the clustering algorithms.

- Although `k=3` for `K-Means` had the highest score, there was a 3.08% relative difference in the Silhouette score for `k=3` and that of `k=4`.
- This difference was the smallest relative to that of the other pairs.
- Thus, the data could be grouped into 3 or 4 clusters, but each algorithm performed the best with 3 clusters.
- When viewing the clusters with respect to the `seven days of the week`, the number of trips varied, but the trips were grouped through similarities found in `groups of weekdays` and `groups of weekends` across both `DBSCAN` and `K-Means`.
- The results of both algorithms indicated that the clusters were likely overlapping, which may have inherently lowered the Silhouette scores.
- *Dimensionality Reduction* through `PCA` and `t-SNE` provided the best 2D projections of the `K-Means (k=3)` clusters.
    - Through these two techniques, all 3 clusters were visibly distinguishable.

Other Considerations

Sources

1. Kaggle Datasets
   a. https://www.kaggle.com/pranavbadami/nj-transit-amtrak-nec-performance
2. Decorative Pictures (train traveling on railroad)
   a. https://pixabay.com/