

CS 439 25F

Clustering: K-Means and t-SNE

...

Group 24

Menyhert [amm926]

Rida Mohammad [rm1724]

Kaushik Murali [km1526]

Project Description

- This project explores clustering patterns in New York City yellow taxi rides using K-Means and t-SNE.
- The objective is to identify natural groupings of trips based on quantitative features such as trip distance, fare amount, tip amount, pickup/drop-off locations, and time of day.
- After preprocessing and feature scaling, K-Means is applied to form clusters that may represent typical ride types (e.g., short local trips, airport runs, high-tip rides).
- t-SNE (t-Distributed Stochastic Neighbor Embedding) is used to reduce dimensionality for 2-D visualization of the cluster structure.
- The project analyzes how temporal and spatial factors affect how people use taxis and how much they spend.

What t-SNE Does and Why We Used It

What t-SNE Does

- Reduces high-dimensional data down to 2D
- Places similar points close together and separates different points
- Preserves the local structure of the data, meaning similar trips stay grouped

Why We Used It

- Our dataset has many features (distance, fare, tips, time, locations) that interact in complex ways
- Linear methods like PCA couldn't show meaningful separation (as we'll see)
- t-SNE captures non-linear patterns and reveals groups that other methods hide

How It Works (simple)

- Finds which trips are most similar in the original dataset
- Builds a 2D layout that tries to keep those relationships intact
- Spreads out the data so clusters appear clearly on a map

What We Learned

- Airport trips, micro-trips, and high-passenger rides separate cleanly
- Everyday Manhattan rides blend into one large cluster
- t-SNE confirmed that our K-Means clusters reflect real structure in the data

Dataset Description and Sample Snapshots

Dataset Name: NYC Yellow Taxi Trip Records - January 2025

Source : NYC Taxi and Limousine Commission (TLC)

[TLC Trip Record Data](#)

Size and Format

- CSV format, ~90 MB (\approx 850K trip records)
- Average Entry Size: ~114 bytes per record
- Structure: One row per trip; 20 total columns
- Each record represents one completed yellow taxi trip

Key Columns

- Categorical identifiers - *VendorID*, *payment_type*
- Trip timestamps - *tpep_pickup_datetime*, *tpep_dropoff_datetime*
- Quantitative features - *passenger_count*, *trip_distance*, *fare_amount*, *tip_amount*, *total_amount*
- Spatial identifiers for pickup and drop-off zones - *PULocationID*, *DOLocationID*
- Pricing modifiers - *RatecodeID*, *tolls_amount*, *congestion_surcharge*, *Airport_fee*

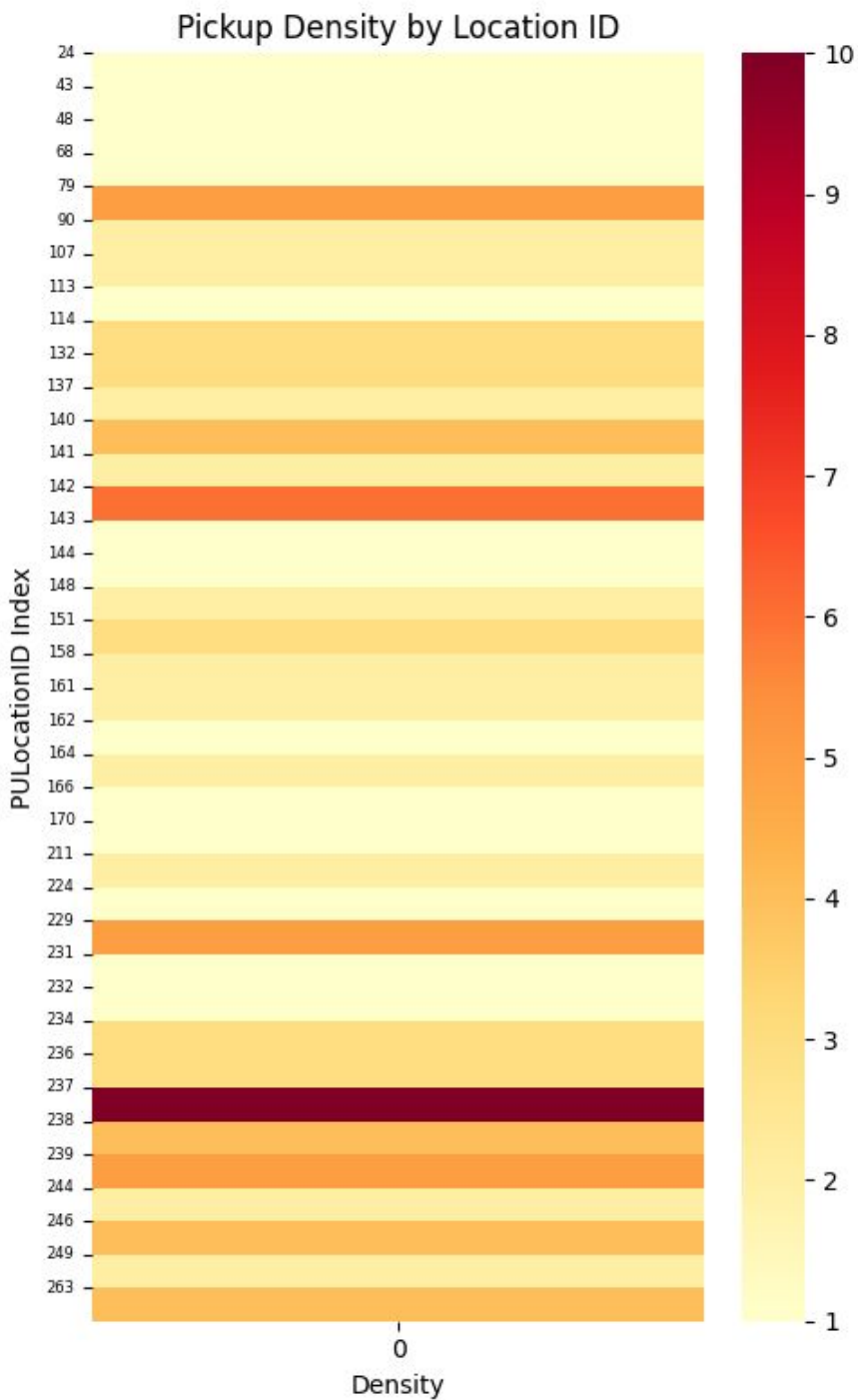
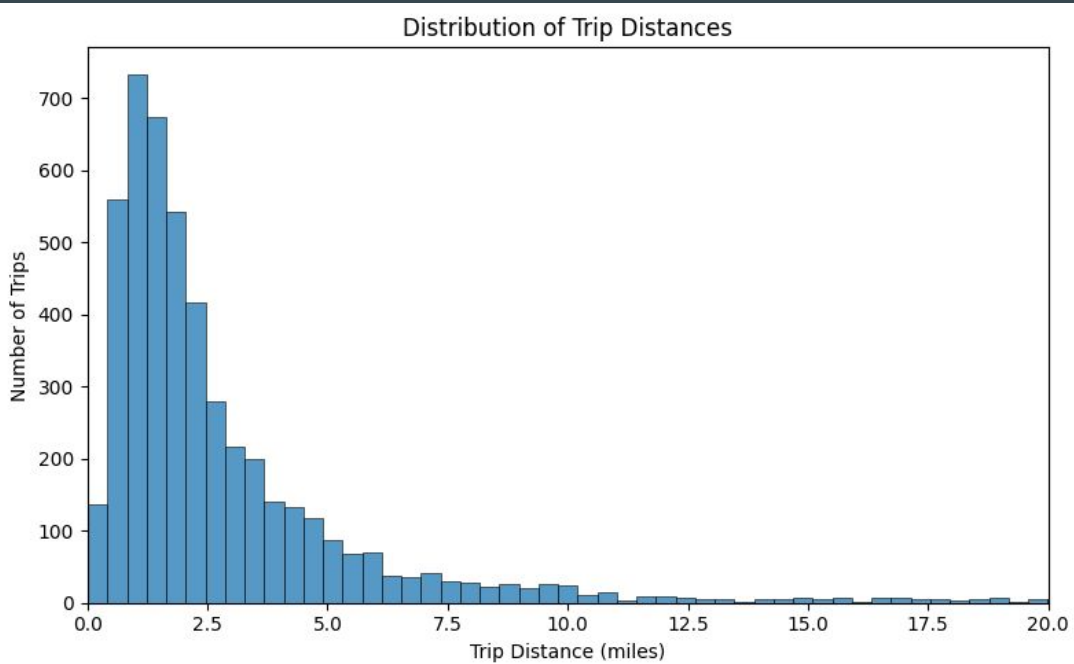
Feature Engineering for Clustering

- Extracted *pickup_hour* and *pickup_weekday* from timestamps
- Filtered out trips with missing or zero distance/fare values
- Scaled numeric features with *StandardScaler* for uniform variance

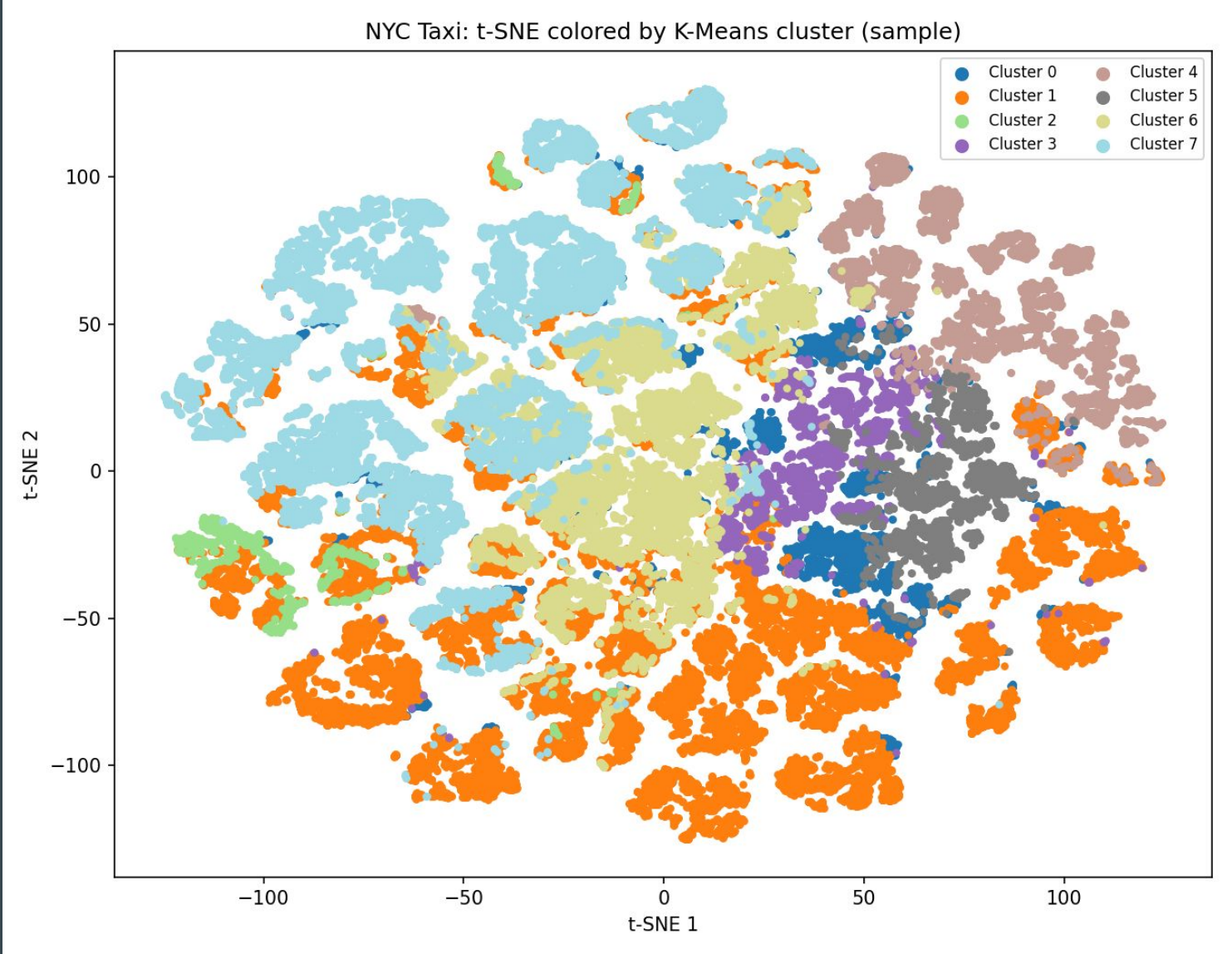
Dataset Snapshot

	VendorID	1	tpet_pickup_datetime	tpet_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	payment_type	fare_amount	extra
1	2		2025-01-01 00:14:27	2025-01-01 00:20:01	3.0	0.52	1.0	N	244	244	2	7.2	1.0
2	2		2025-01-01 00:21:34	2025-01-01 00:25:06	3.0	0.66	1.0	N	244	116	2	5.8	1.0
3	2		2025-01-01 00:48:24	2025-01-01 01:08:26	2.0	2.63	1.0	N	239	68	2	19.1	1.0
4	2		2025-01-01 00:00:02	2025-01-01 00:09:36	1.0	1.71	1.0	N	237	262	2	11.4	1.0
5	2		2025-01-01 00:20:28	2025-01-01 00:28:04	1.0	2.29	1.0	N	237	75	2	11.4	1.0
6	2		2025-01-01 00:33:58	2025-01-01 00:37:23	1.0	0.56	1.0	N	263	236	1	5.8	1.0
7	2		2025-01-01 00:42:40	2025-01-01 00:55:38	3.0	1.99	1.0	N	236	151	2	14.2	1.0
8	2		2025-01-01 00:01:41	2025-01-01 00:07:14	1.0	0.71	1.0	N	79	107	2	-7.2	-1.0
9	2		2025-01-01 00:01:41	2025-01-01 00:07:14	1.0	0.71	1.0	N	79	107	2	7.2	1.0
10	2		2025-01-01 00:05:49	2025-01-01 00:20:00	2.0	3.45	1.0	N	263	107	1	17.7	1.0
11	2		2025-01-01 00:34:40	2025-01-01 00:51:19	2.0	1.19	1.0	N	246	170	1	14.9	1.0
12	2		2025-01-01 00:55:54	2025-01-01 01:00:38	1.0	0.69	1.0	N	137	233	4	-6.5	-1.0
13	2		2025-01-01 00:55:54	2025-01-01 01:00:38	1.0	0.69	1.0	N	137	233	4	6.5	1.0
14	2		2025-01-01 00:14:14	2025-01-01 00:22:46	2.0	1.88	1.0	N	113	90	1	11.4	1.0
15	2		2025-01-01 00:32:01	2025-01-01 00:46:43	1.0	3.06	1.0	N	90	246	1	17.7	1.0
16	2		2025-01-01 00:00:37	2025-01-01 00:04:46	1.0	0.86	1.0	N	140	263	2	6.5	1.0
17	2		2025-01-01 00:11:07	2025-01-01 00:23:29	3.0	3.98	1.0	N	140	79	1	19.1	1.0
18	2		2025-01-01 00:32:47	2025-01-01 01:01:49	1.0	4.41	1.0	N	246	236	1	28.9	1.0
19	2		2025-01-01 00:32:04	2025-01-01 00:40:54	1.0	1.51	1.0	N	237	140	1	10.7	1.0
20	2		2025-01-01 00:53:47	2025-01-01 00:58:02	1.0	0.65	1.0	N	263	141	1	5.8	1.0
21	2		2025-01-01 00:16:55	2025-01-01 00:27:36	1.0	1.66	1.0	N	238	236	1	12.1	1.0
22	2		2025-01-01 00:39:33	2025-01-01 00:43:53	1.0	0.77	1.0	N	239	142	1	6.5	1.0
23	2		2025-01-01 00:59:29	2025-01-01 01:27:47	1.0	4.26	1.0	N	239	113	1	27.5	1.0
24	2		2025-01-01 00:15:41	2025-01-01 01:03:03	4.0	3.05	1.0	N	114	161	1	37.3	1.0
25	2		2025-01-01 00:39:26	2025-01-01 00:57:14	1.0	2.31	1.0	N	224	68	1	17.0	1.0
26	2		2025-01-01 00:07:07	2025-01-01 00:13:18	2.0	1.02	1.0	N	237	236	1	7.9	1.0
27	2		2025-01-01 00:22:56	2025-01-01 00:33:12	2.0	1.5	1.0	N	237	140	1	10.7	1.0
28	2		2025-01-01 00:43:44	2025-01-01 01:05:18	4.0	2.26	1.0	N	237	238	1	19.8	1.0
29	2		2025-01-01 00:52:53	2025-01-01 00:59:31	1.0	1.63	1.0	N	237	263	1	9.3	1.0
30	2		2025-01-01 00:37:14	2025-01-01 00:42:58	4.0	1.01	1.0	N	107	170	1	7.9	1.0
31	2		2025-01-01 00:46:47	2025-01-01 01:02:28	1.0	1.58	1.0	N	137	79	2	12.1	1.0
32	2		2025-01-01 00:19:52	2025-01-01 00:36:32	1.0	3.41	1.0	N	232	49	1	19.8	1.0
33	2		2025-01-01 00:25:03	2025-01-01 00:54:59	2.0	5.17	1.0	N	211	263	1	28.9	1.0
34	2		2025-01-01 00:14:46	2025-01-01 00:37:07	1.0	6.06	1.0	N	79	260	2	28.2	1.0
35	2		2025-01-01 00:11:10	2025-01-01 00:26:43	1.0	1.63	1.0	N	246	48	1	14.9	1.0
36	2		2025-01-01 00:30:40	2025-01-01 00:49:16	1.0	2.22	1.0	N	48	239	1	17.0	1.0
37	2		2025-01-01 00:52:50	2025-01-01 01:04:50	1.0	1.26	1.0	N	142	48	1	12.1	1.0
38	2		2025-01-01 00:48:18	2025-01-01 01:03:53	1.0	4.2	1.0	N	166	48	1	21.2	1.0
39	2		2025-01-01 00:36:56	2025-01-01 00:53:45	1.0	3.0	1.0	N	151	140	1	18.4	1.0
40	2		2025-01-01 00:11:13	2025-01-01 00:21:35	2.0	2.0	1.0	N	249	246	1	12.8	1.0
41	2		2025-01-01 00:19:18	2025-01-01 00:31:06	1.0	2.2	1.0	N	238	237	1	14.2	1.0
42	2		2025-01-01 00:33:54	2025-01-01 01:04:28	2.0	3.46	1.0	N	237	249	1	26.8	1.0
43	2		2025-01-01 00:09:04	2025-01-01 00:18:09	1.0	1.9	1.0	N	140	162	1	11.4	1.0

Dataset Graphs



Clusters

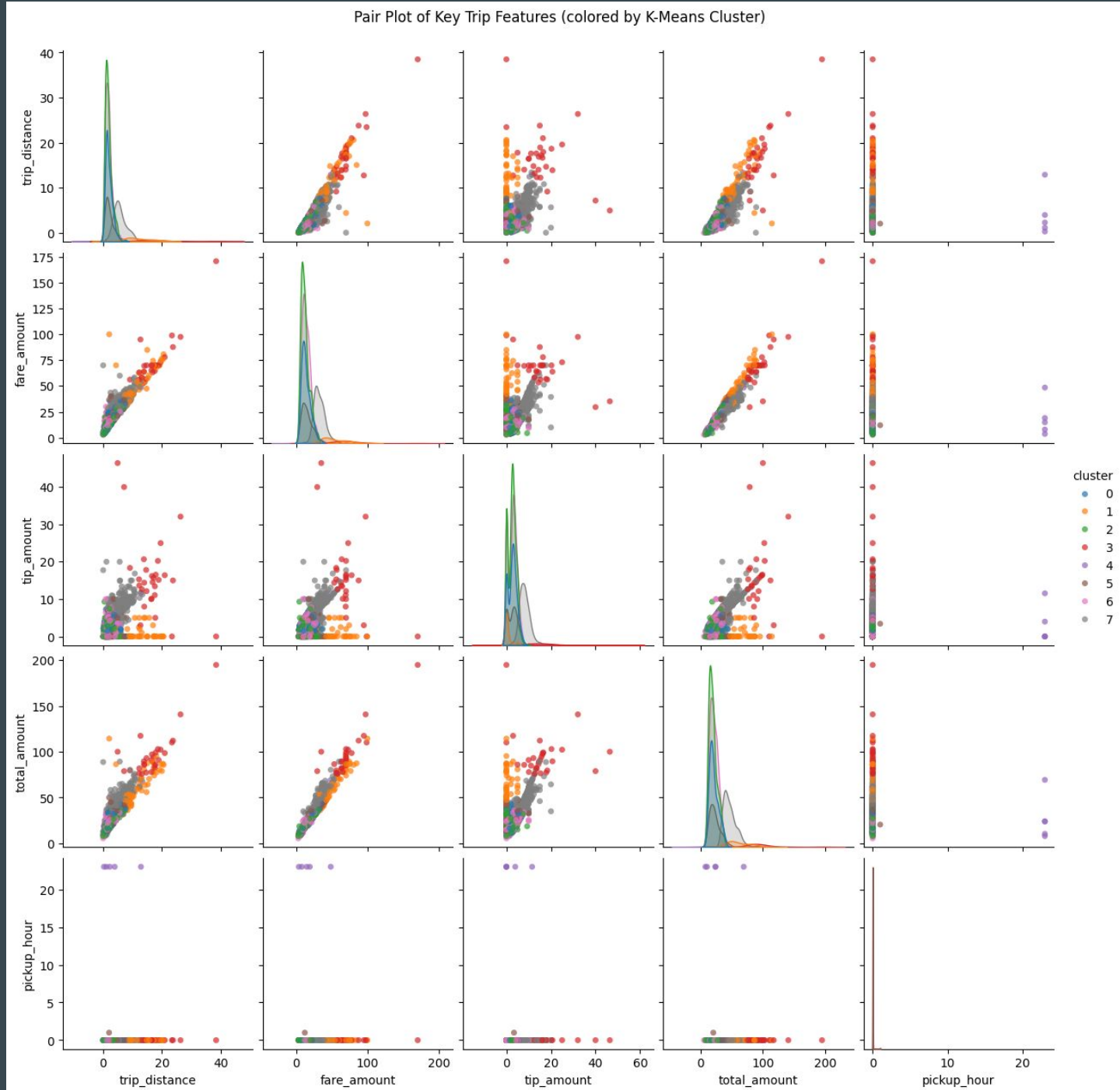


	passenger_count	trip_distance	fare_amount	tip_amount	total_amount	PULocationID	DOLocationID	pickup_hour	pickup_weekday
cluster									
0	1.25	8.47	39.04	9.89	59.53	147.54	148.38	14.66	2.99
1	1.16	2.22	12.49	2.34	19.22	158.84	149.93	5.55	2.94
2	1.17	1.75	11.70	2.45	18.71	206.50	174.05	15.38	4.54
3	1.47	14.33	63.89	1.04	74.39	140.29	140.93	13.43	2.98
4	3.78	2.29	13.55	2.52	20.79	167.73	166.20	14.95	3.36
5	1.62	17.60	75.98	16.76	104.02	141.01	150.34	14.34	3.08
6	1.16	2.07	12.56	2.42	19.81	96.80	132.82	16.40	2.99
7	1.13	1.69	11.44	2.56	19.11	200.52	200.91	15.55	1.52

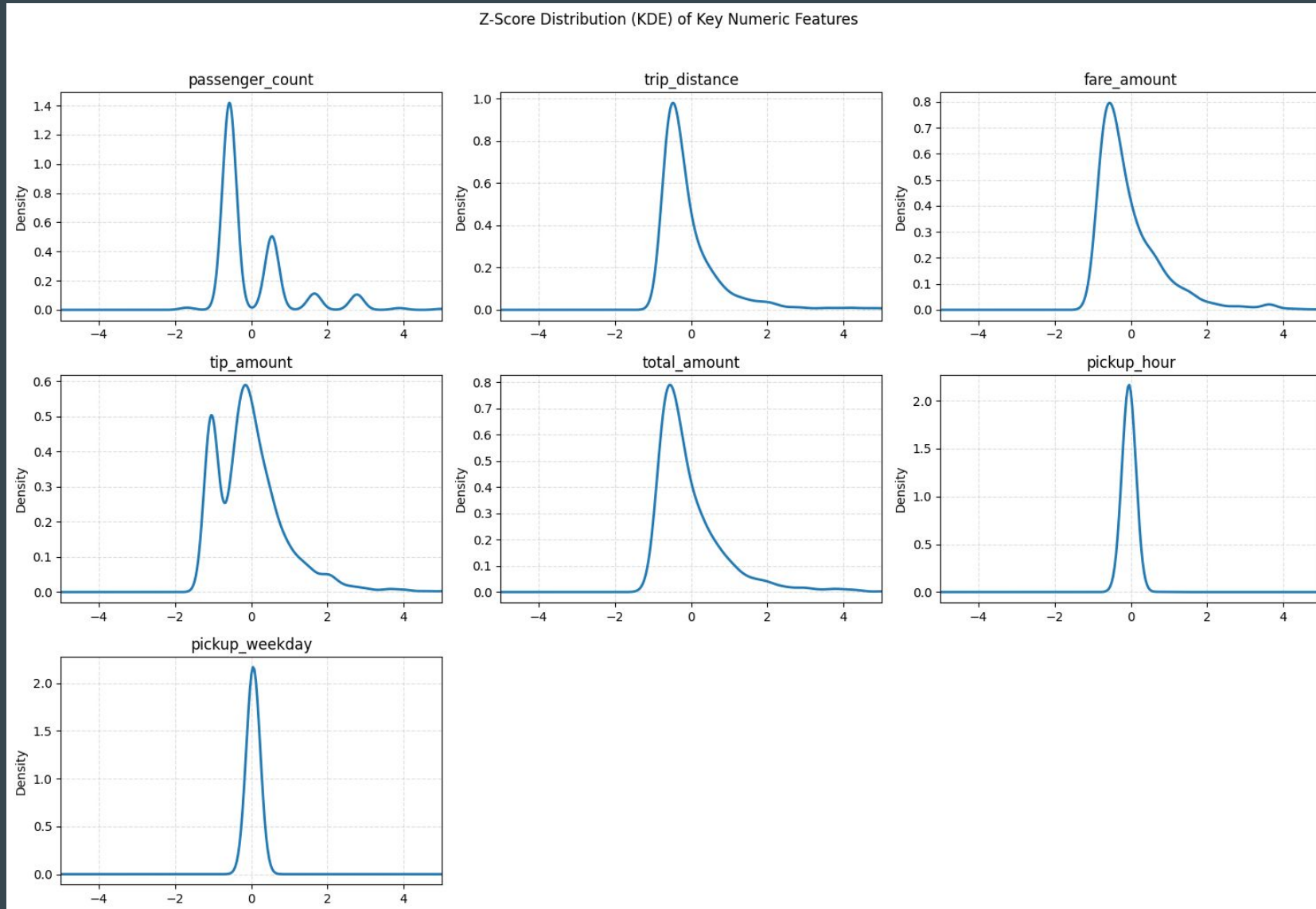
Analyzing the Clusters

- 0: Short Midtown locals. Low distance, average tip percent ~23%.
- 1: Airports. 46% JFK, 22% LaGuardia. Long trips, highest fares and tips.
- 2: Downtown/East-side locals. Clinton East, Penn, East Village heavy. Medium distance.
- 3: Midtown/Chelsea/Central Park corridor. Short trips, big volume.
- 4: Upper East/West Side locals. UES/UWS dominate. Very short trips, lower fares.
- 5: Midtown mix including Penn, Times Sq, Murray Hill. Slightly longer than 0 & 4.
- 6: High passenger counts (mean ~3.8). Likely shared rides, shuttles, or data entry artifacts.
- 7: Outlier cluster. Size = 3.

Pair Plot



Z-Score Distribution



Questions to be addressed

- Can NYC taxi trips be clustered into distinct ride types?
- Which trip features most strongly define these clusters?
- What number of clusters best fits the data?
- How do ride patterns differ across times and days?
- How effectively does t-SNE visualize cluster separation?

Data Transformation Flowchart

1. Data Import

- Load yellow_tripdata_2025-01.csv (≈90 MB) into a pandas DataFrame.

2. Data Cleaning

- Remove rows with missing or invalid values (zero fare, zero distance).
- Keep only numeric and relevant categorical fields (drop nonpositive distances/fares).

3. Feature Engineering

- Extract pickup_hour and pickup_weekday from tpep_pickup_datetime.
- Compute derived metrics such as $\text{tip_percentage} = \text{tip_amount} / \text{fare_amount}$.

4. Feature Selection

- Choose features for clustering: ['passenger_count', 'trip_distance', 'fare_amount', 'tip_amount', 'total_amount', 'PULocationID', 'DOLocationID', 'pickup_hour', 'pickup_weekday']

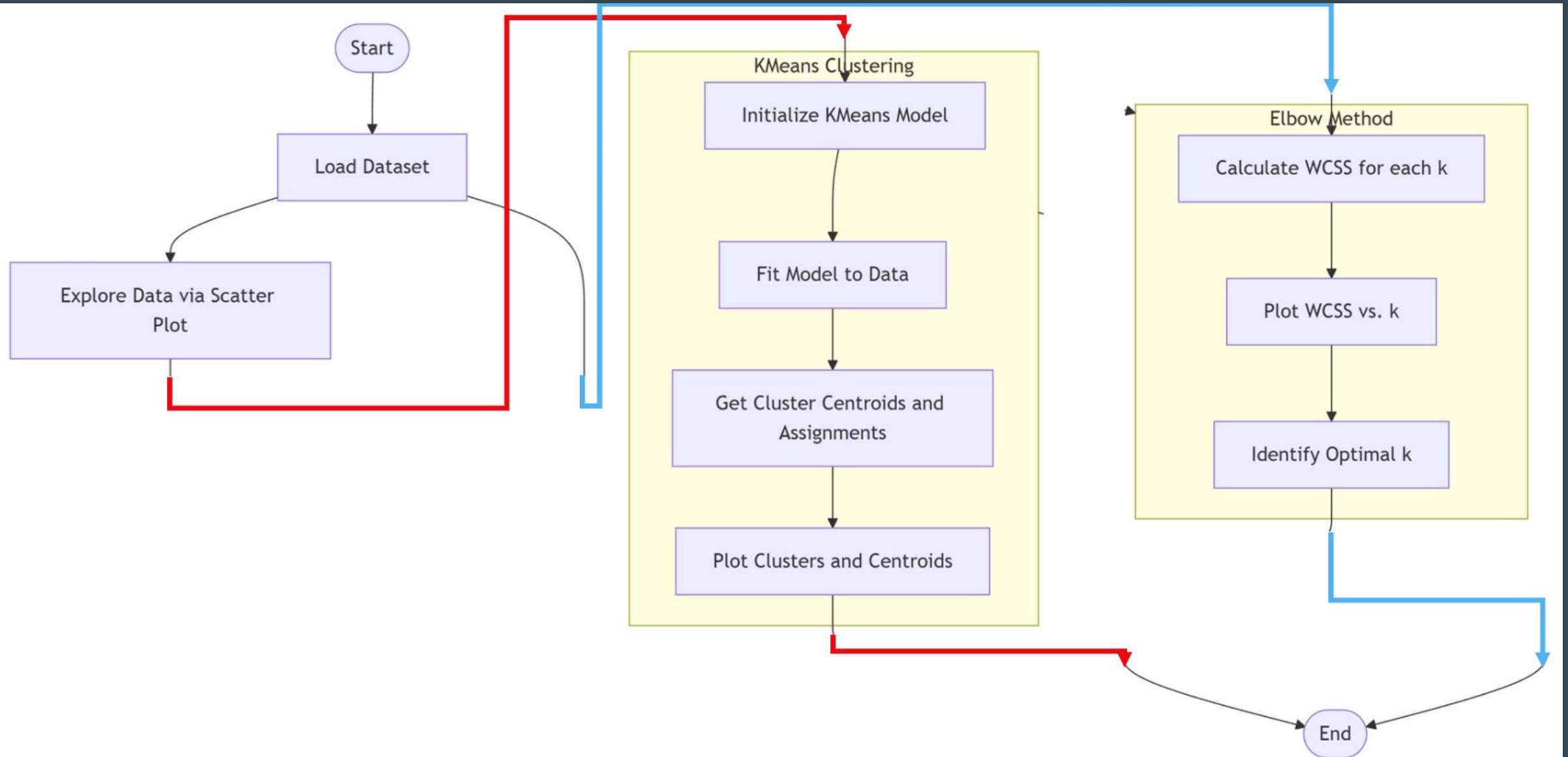
5. Scaling

- Standardize all numeric columns using StandardScaler.

6. Clustering and Visualization

- Apply K-Means for grouping.
- Use t-SNE for 2-D visualization of clusters.

Flowchart



Exploration Algorithms Used

Data Cleaning

- Removed invalid records (zero or negative distance/fare).
- Dropped missing values in key columns.
- Filtered out unrealistic outliers in trip distance.

Feature Engineering

- Extracted *pickup_hour* and *pickup_weekday* from timestamps.
- Created $\text{tip_percentage} = \text{tip_amount} / \text{fare_amount}$ for better spending analysis.

Clustering

- K-Means Algorithm
 - Applied K-Means to group trips by numeric similarity.
 - Determined optimal k using the elbow method.

Dimensionality Reduction

- t-SNE (t-Distributed Stochastic Neighbor Embedding)
 - Applied t-SNE to project features into 2D.
 - Preserved local structure for visual cluster separation.

Visualization

- 2D scatter plots colored by cluster label.
- Density maps showing pickup location and time distributions.

Computational Platform and Software Libraries Used

Programming Language:

- Python 3.11

Development Environment:

- Jupyter Notebook (Anaconda distribution)

Core Libraries:

- pandas: data loading, cleaning, feature engineering
- numpy: numerical operations
- scikit-learn: clustering (K-Means), dimensionality reduction (t-SNE), scaling, PCA
- matplotlib, seaborn: visualization
- Plotly Dash, webbrowser, threading: dashboard creation

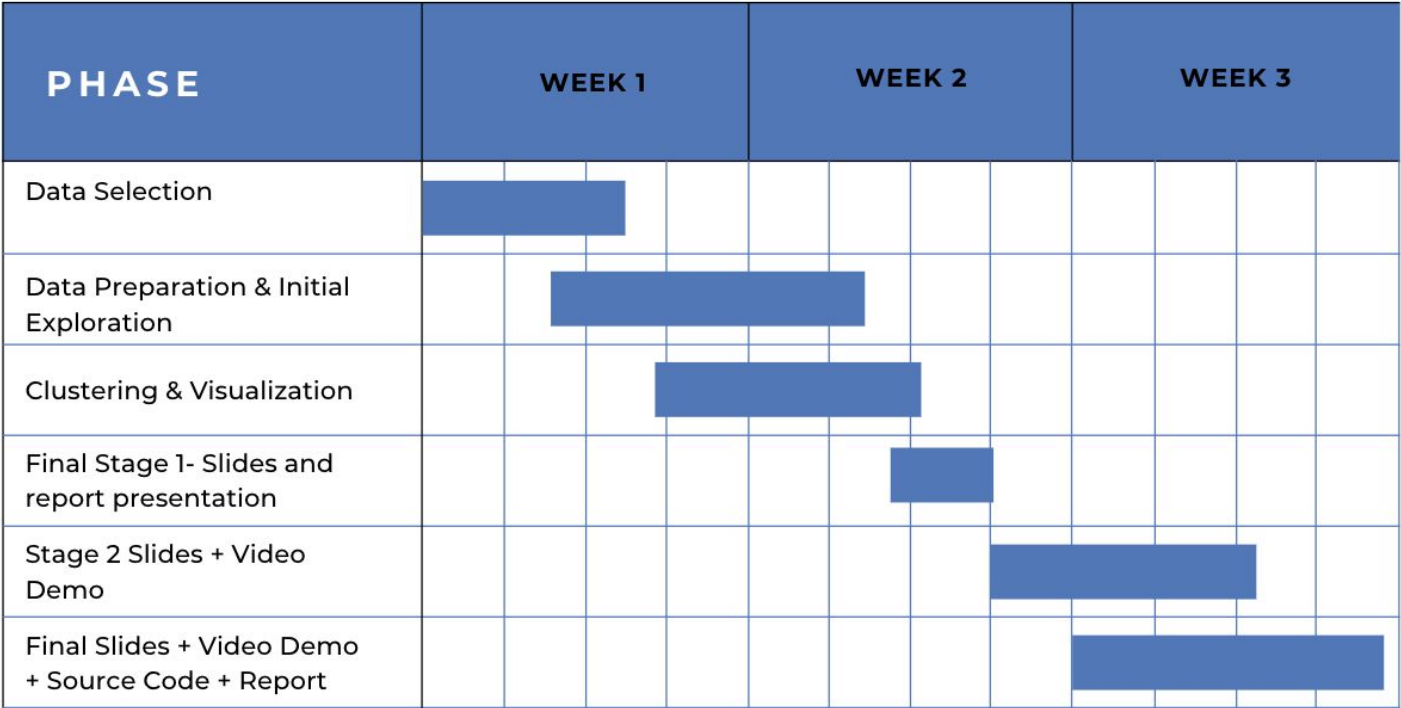
Hardware Used:

- Local machine (Windows 11, 32GB RAM)
- Dataset size \approx 90 MB

Gantt Chart

Phase	Tasks	Deliverables / Due Date
Stage 1 - Data Preparation & Initial Exploration	Load dataset, clean data, engineer features (pickup hour, weekday), verify dataset size ≥ 8 MB	Week 10 Mon - Stage 1 Slides
Stage 2 - Clustering & Visualization	Apply K-Means and t-SNE, tune k, generate visualizations (histogram, scatter, geographic heatmap), interpret clusters	Week 14 Mon - Stage 2 Slides + Video Demo
Final Stage - Full Integration & Submission	Refine visuals and explanations, finalize code and written report (LaTeX format), record final demo	Week 15 Wed - Final Slides + Video + Source Code + Report

Clustering: K-Means and t-SNE Project Report 1



Division of Labor

Menyhert (amm926)

- Imported and cleaned the NYC Yellow Taxi dataset
- Engineered time-based and tip-related features
- Implemented data scaling, K-Means clustering, and t-SNE visualization
- Developed and debugged the interactive Plotly Dash dashboard
- Designed final graphs and slide layouts; compiled and formatted the presentation

Rida Mohammad (rm1724)

- Researched dataset details and documentation (TLC data dictionary)
- Drafted Dataset Description, Sample Snapshots, and Data Transformation Flowchart slides
- Summarized algorithm theory and Evaluation Techniques content
- Verified cluster quality metrics (Elbow method)

Kaushik Murali (km1526)

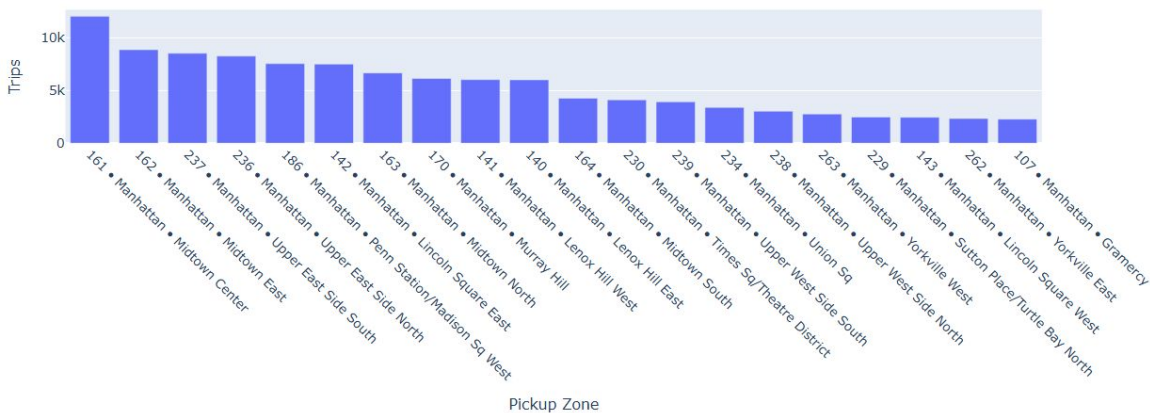
- Assisted with feature selection and outlier filtering
- Helped analyze and interpret cluster patterns (spatial and temporal)
- Prepared visualizations (Pair Plot, Z-Score Distribution)

Interactive Dashboard

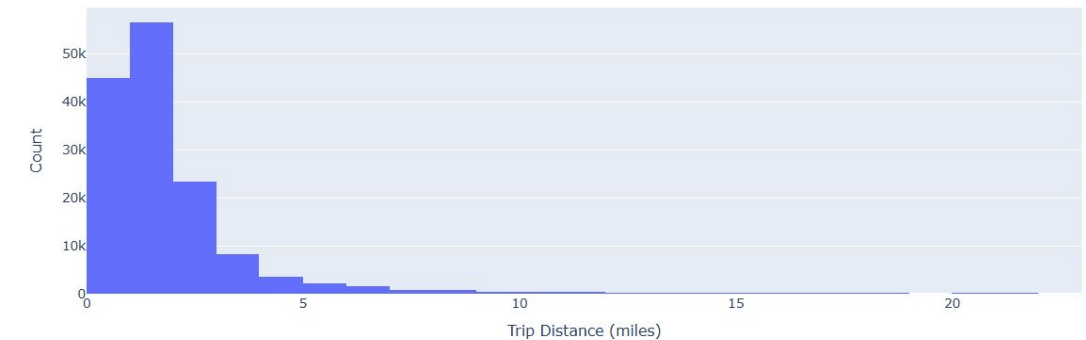
NYC Taxi Clustering Dashboard

Cluster 0

Top Pickup Zones for Cluster 0

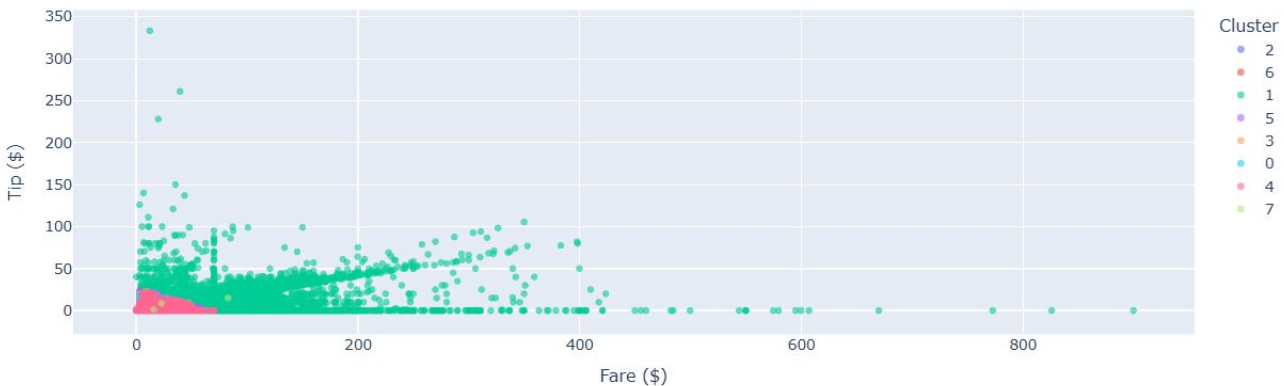


Trip Distance Distribution



Avg Fare: \$12.71 | Avg Tip %: 23.3% | Avg Passengers: 1.12

Fare vs Tip by Cluster



Evaluation Techniques

Cluster Quality Metrics:

- Elbow Method: determines optimal number of clusters ($k = 8$)

Dimensionality Reduction Check:

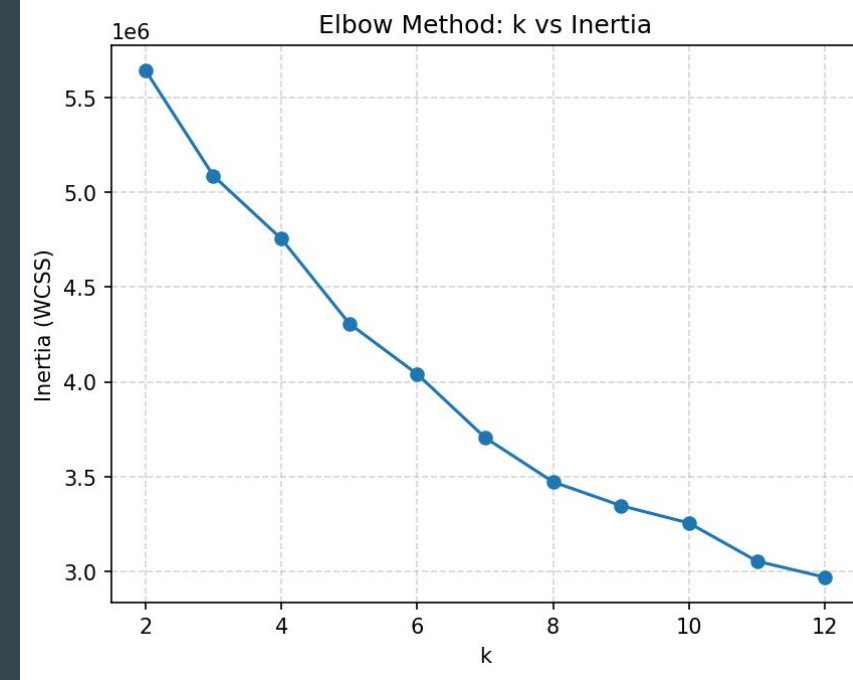
- t-SNE Visualization: validates visual separation of cluster groups

Feature Impact:

- Compared clustering results with and without spatial features (*PULocationID*, *DOLocationID*)

Validation:

- Re-clustered random subsets to test result stability
- Visually inspected cluster density patterns on NYC map



Interesting Findings - Behavioral Patterns

- Airport rides formed a distinctly isolated cluster due to extreme distance and fare values.
- Everyday Manhattan trips merged into one dominant cluster, capturing most rides in the dataset.
- Micro-trips (sub-1 mile) appeared as their own cluster, showing strong neighborhood-driven behavior.
- High-passenger trips separated clearly, indicating shared rides or data irregularities.

Interesting Findings - Technical Insights

- PCA showed little separation, meaning variance is dominated by a few continuous features.
- t-SNE revealed meaningful structure, capturing non-linear relationships between distance, fare, and location.
- Elbow method validated $k \approx 8$, with diminishing returns beyond that point.
- Location density played a large role, with a few pickup zones strongly influencing cluster shapes.

References

New York City Taxi and Limousine Commission (TLC). NYC Yellow Taxi Trip Records, January 2025.

[TLC Trip Record Data](#)

zyBooks Ch 10.2 Unsupervised Learning - K-means clustering.

[zyBooks Ch. 10](#)

Supplemental Material for Lecture 10 : Unsupervised Learning.

[Supplemental Material for Lecture 10](#)

t-SNE clearly explained. Medium - Kemal Erdem

[t-SNE Clearly Explained](#)

[Link to all code/files](#)