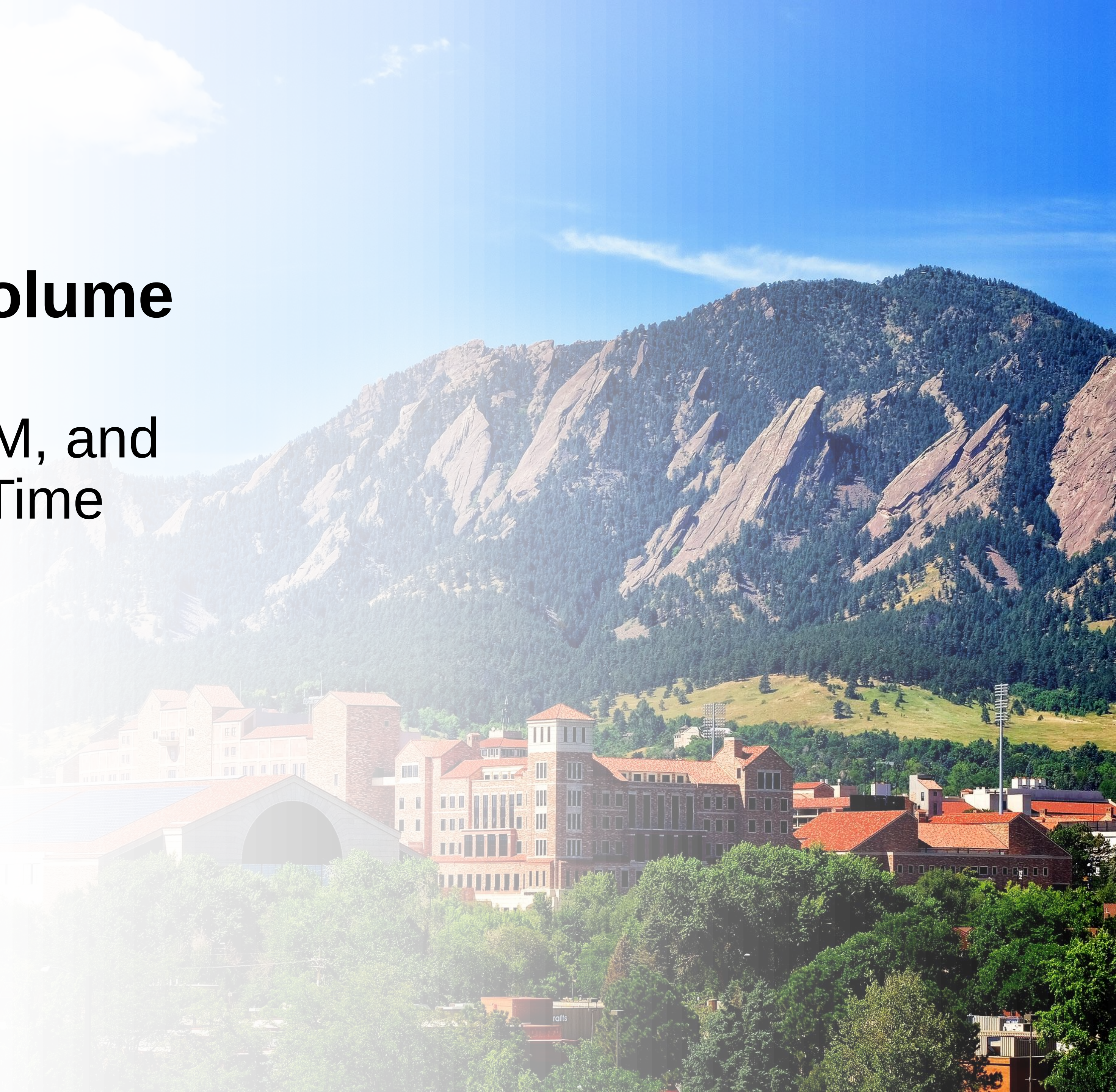
Taxi! Taxi! – NYC Taxi Volume Anomaly Detection

Comparing DBSCAN, OCSVM, and Autoencoder Approaches to Time Series Anomaly Detection

Nick Vastine
DTSA5506 – Data Mining Project
Final Project
April 21, 2024



Data Science
UNIVERSITY OF COLORADO **BOULDER**

Agenda

- **Executive Summary**
- **Problem Area** – Anomaly Detection in Unlabeled Time Series using NYC Taxi Volume to Compare Methods
- **Related Work** – Anomaly Detection Techniques
- **Proposed Work**
  - **Data** – NYC Taxi Volume for 2019
  - **Key Tasks** – Feature engineering, algorithm modeling
  - **Modeling** – DBSCAN, OCSVM, Autoencoders
- **Evaluation** – Outliers, Visualization, Feature Importance
- **Discussion**
- **Conclusion** and **Future Work**

Executive Summary

- Evaluate three anomaly detection methods (DBSCAN, OCSVM, and autoencoder) on unlabeled, time-series NYC Taxi reporting.
- Assess based on hourly- and daily-aggregate datasets with engineered features (rolling average, rate of change, cyclic).
- All models performed similarly on simpler daily-aggregate data, while OCSVM was much easier to interpret for hour-aggregate.
  - In hour-aggregate, OCSVM appeared to label entire days as anomalous, while other methods focused on hourly variation.
- Difference in methods qualified in PCA/t-SNE visualizations and feature importance from Random Forest classifier confirmed daily-anomaly similarities and hour-anomaly differences.

Problem Area – Anomaly Detection

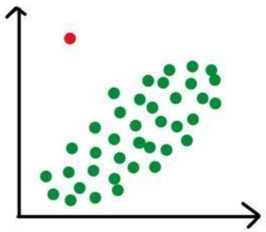
- Anomaly detection applies to a wide range of applications.

Cybersecurity
Suspicious Network Activity

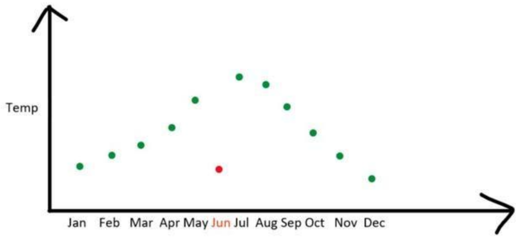
Industry
Sensor/Machinery Fault

Finance
Fraudulent Transactions

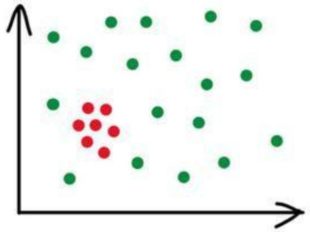
- There are several types of anomalies to detect:



Global
Differs from all points



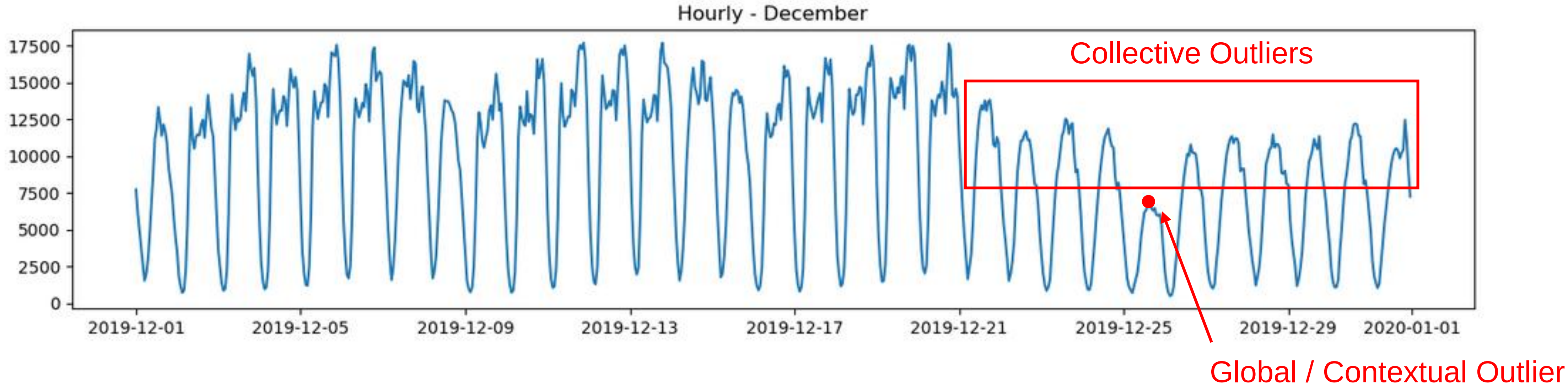
Contextual
Depends on surrounding points



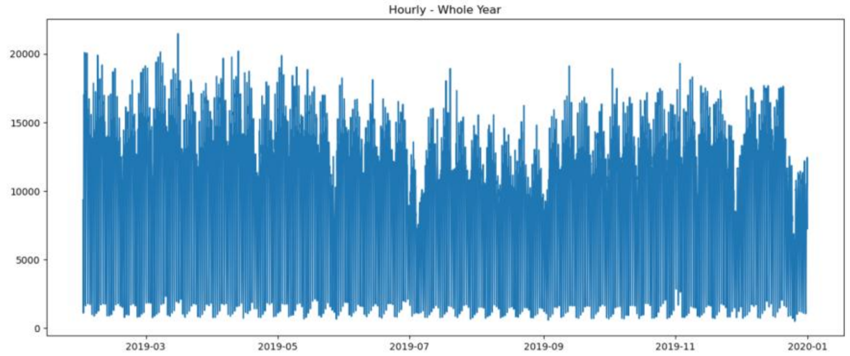
Collective
Group of Outliers

Problem Area – Time Series Data

- Anomaly detection in time-series data has unique challenges
  - Determine ‘normal’ conditions, which can change over time
  - Anomaly detection models may require engineered features to capture time-context, unlike RNN methods
  - Consider interpretability of model (especially neural networks)



Problem Area – Applied Methodology



- NYC Taxi Reporting
  - Identify anomalies in hourly trip volume reported by NYC Yellow Cabs in 2019.
  - Widely applicable business analysis to understand service demand across days, months, and seasons
- Unlabeled time series data with expected anomalies
  - Daily, weekly, monthly, seasonal change
  - Holidays, events, weather effects

Related Work - Methods

- Anomaly detection has many methods and approaches:
  - *Supervised classification* – Using labeled training data to classify anomaly vs. normal.
  - *Unsupervised clustering* – Group similar data points into clusters except for dramatic outliers or anomalies.
  - *Semi-supervised one-class classification* – Train a classification method on normal data, which then can distinguish deviations.
  - *Neural network* – Consider autoencoder structure trained on normal data which reconstructs anomalies poorly, distinguishing outliers.
- Data streams rarely label anomalies directly. The method must learn normal conditions on its own to robustly distinguish anomalies.

Proposed Work

- Compare anomaly detection methodologies applied to unlabeled, time-series NYC taxi data

1. **Data Preparation** – Data sourcing, processing, warehousing
2. **Exploratory Analysis** – Understand data distribution, trends, identifying statistical outliers
3. **Feature Engineering** – Capturing time-series dependencies
4. **Modeling –** Evaluating three methods: Semi-supervised *OCSVM,* Unsupervised *DBSCAN*, Neural Network *Autoencoder*
5. **Evaluation** – Statistical outliers, qualitative assessment
6. **Feature Importance** – Understanding time-series feature value

Proposed Work – Data Preparation

- 84.4M trips recorded in 2019 NYC Taxi Reporting.

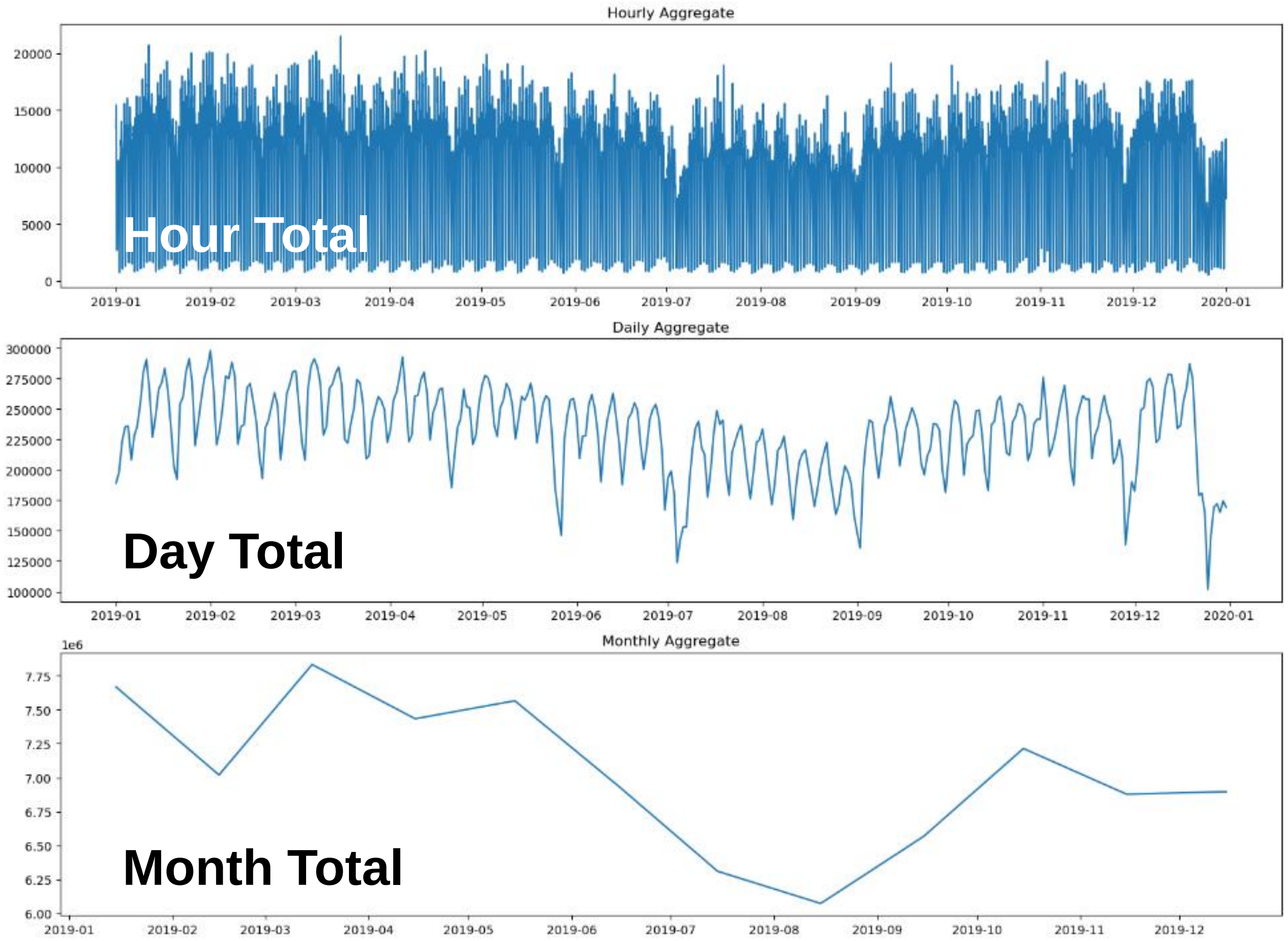
	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	paym
0	1	01/01/2019 12:46:40 AM	01/01/2019 12:53:20 AM	1	1.5	1	N	151	239	

- 8GB file was processed in PostgreSQL to 1) bin entries by hour throughout the year and 2) decompose date into components

	month	day	hour	count	weekday	season	datetime
0	1	1	0	13428	2	0	2019-01-01 00:00:00
1	1	1	1	15444	2	0	2019-01-01 01:00:00
2	1	1	2	13247	2	0	2019-01-01 02:00:00

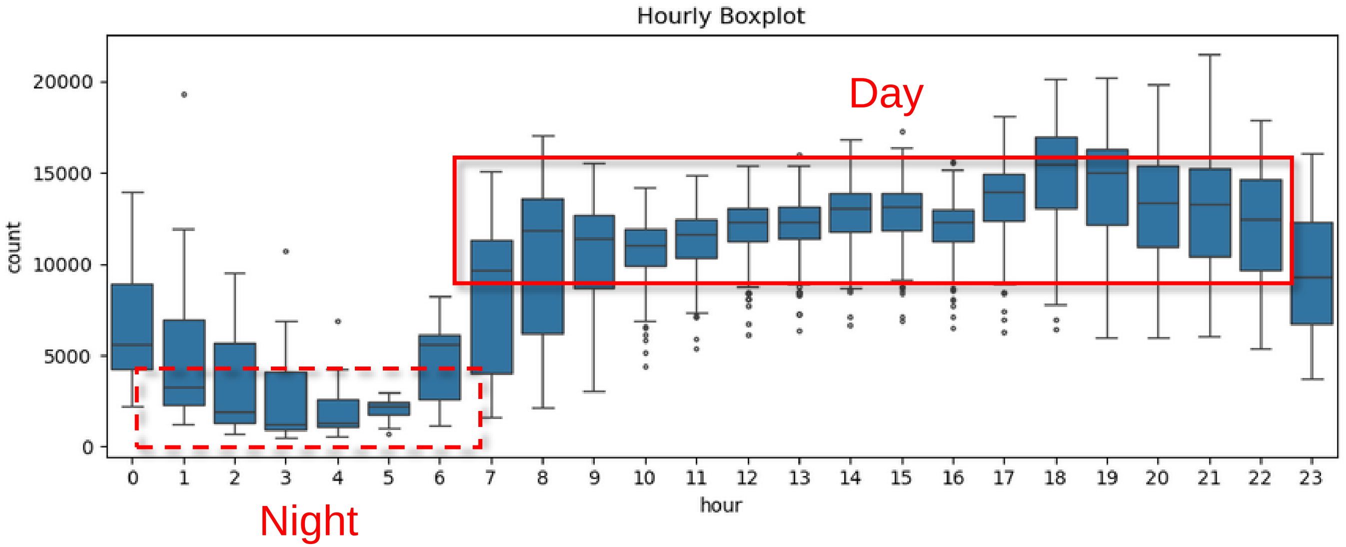
- Reduced dataset to 8760 points for further processing

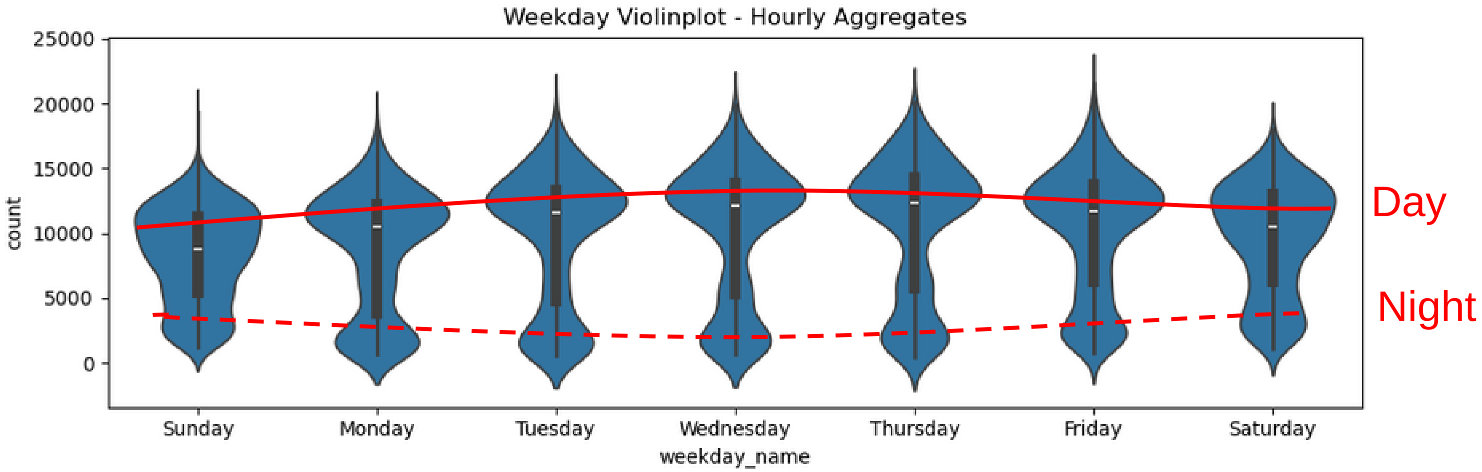
Proposed Work – Exploratory Analysis



- Reduced dataset was explored in Python.
- Binning the hourly aggregates by day and by month reveals larger trends.
    - Lower volume in summer months, weekends, and at night.

Proposed Work – Distribution Analysis



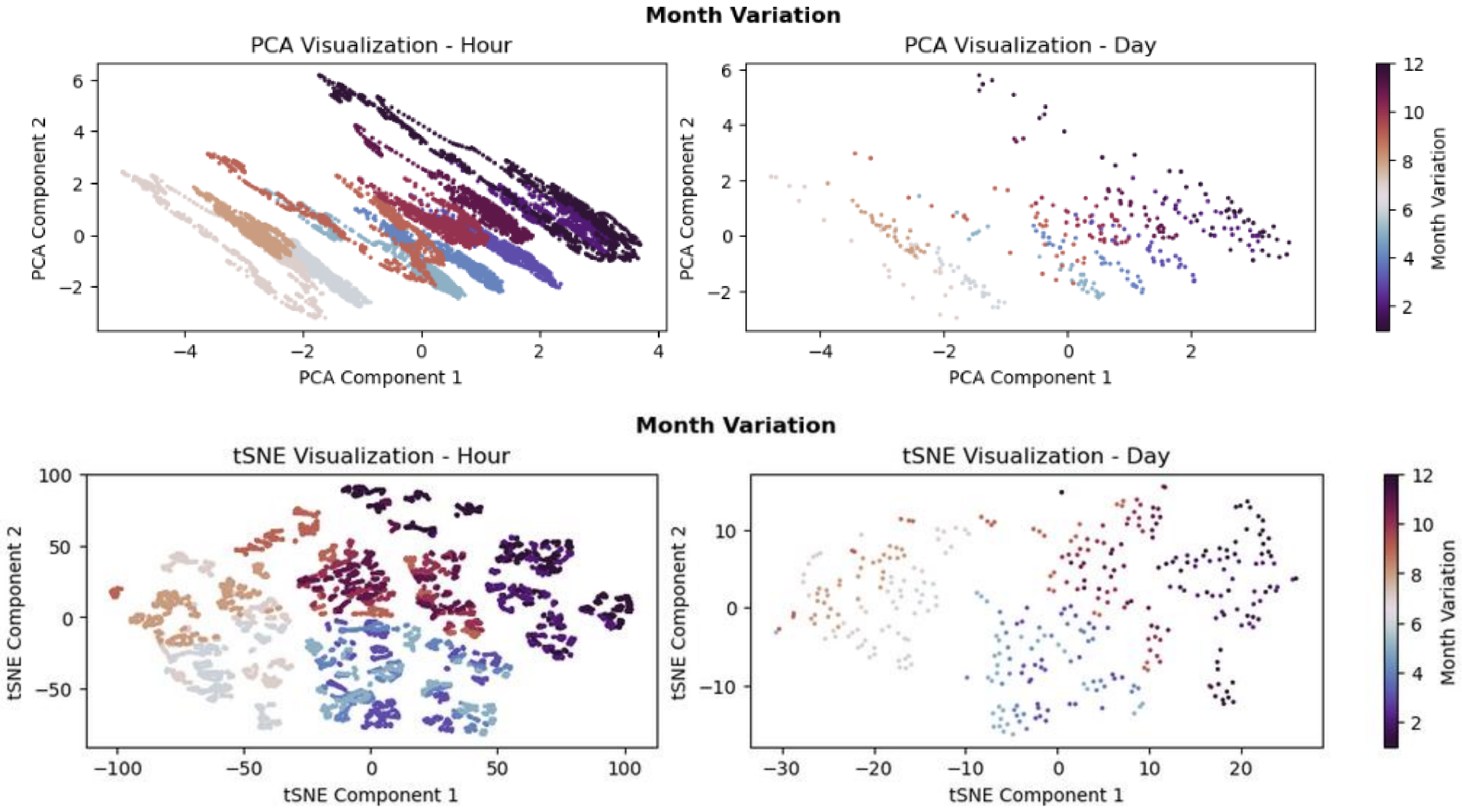


- Hourly boxplot:
  - Decrease in volume in early hours
  - Wide variance during commutes
  - Used to inform statistical outliers
- Weekday violinplot:
  - Lower average on weekends
  - Clearer bi-modal pattern on weekdays

Proposed Work – Feature Engineering

- Features are engineered to preserve time-series context
  - **Rolling Averages** smooth rapid change
  - **Rate of Change** emphasize rapid change
  - **Cyclic Encoding** preserve relative position
    - Ex. 7 AM on Wednesday is encoded with same hour cycle and weekday cycle value to label similar points.

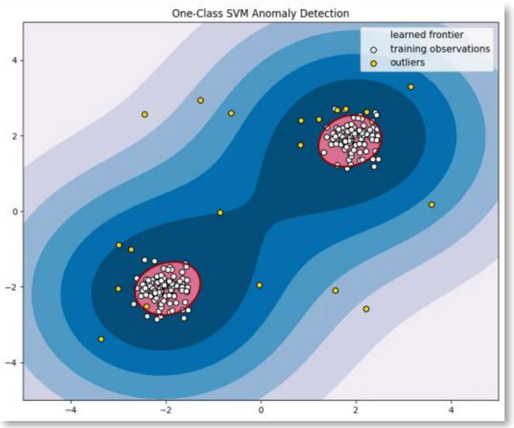
Proposed Work – Pre-Visualization



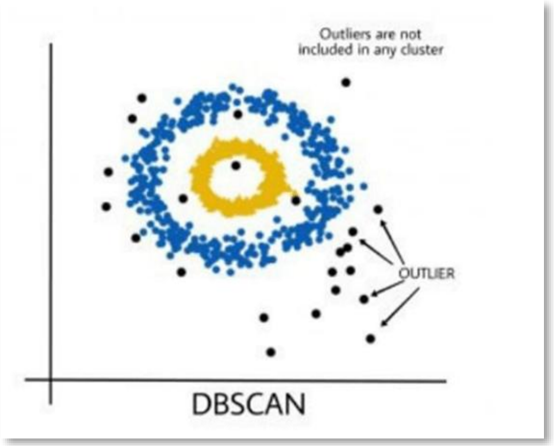
- PCA and t-SNE help visualize variation across reduced dimensions.
- We can use these plots to visualize an algorithm's anomaly approach.

Proposed Work – Modeling

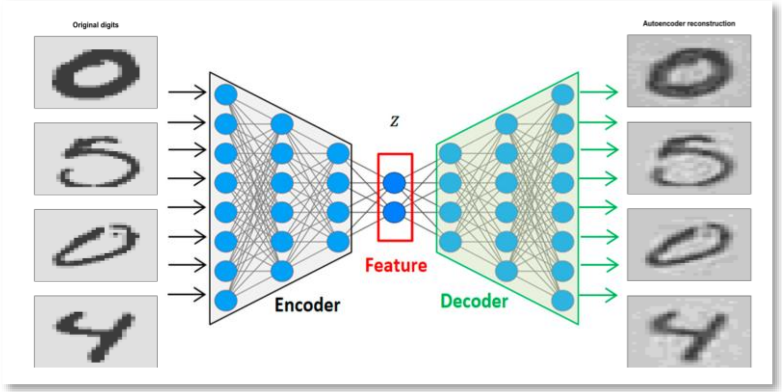
- Evaluate the engineered time-series features and binned, hourly taxi volume using three anomaly detection methods:



OneClassSVM
Train on normal data to develop decision boundary.

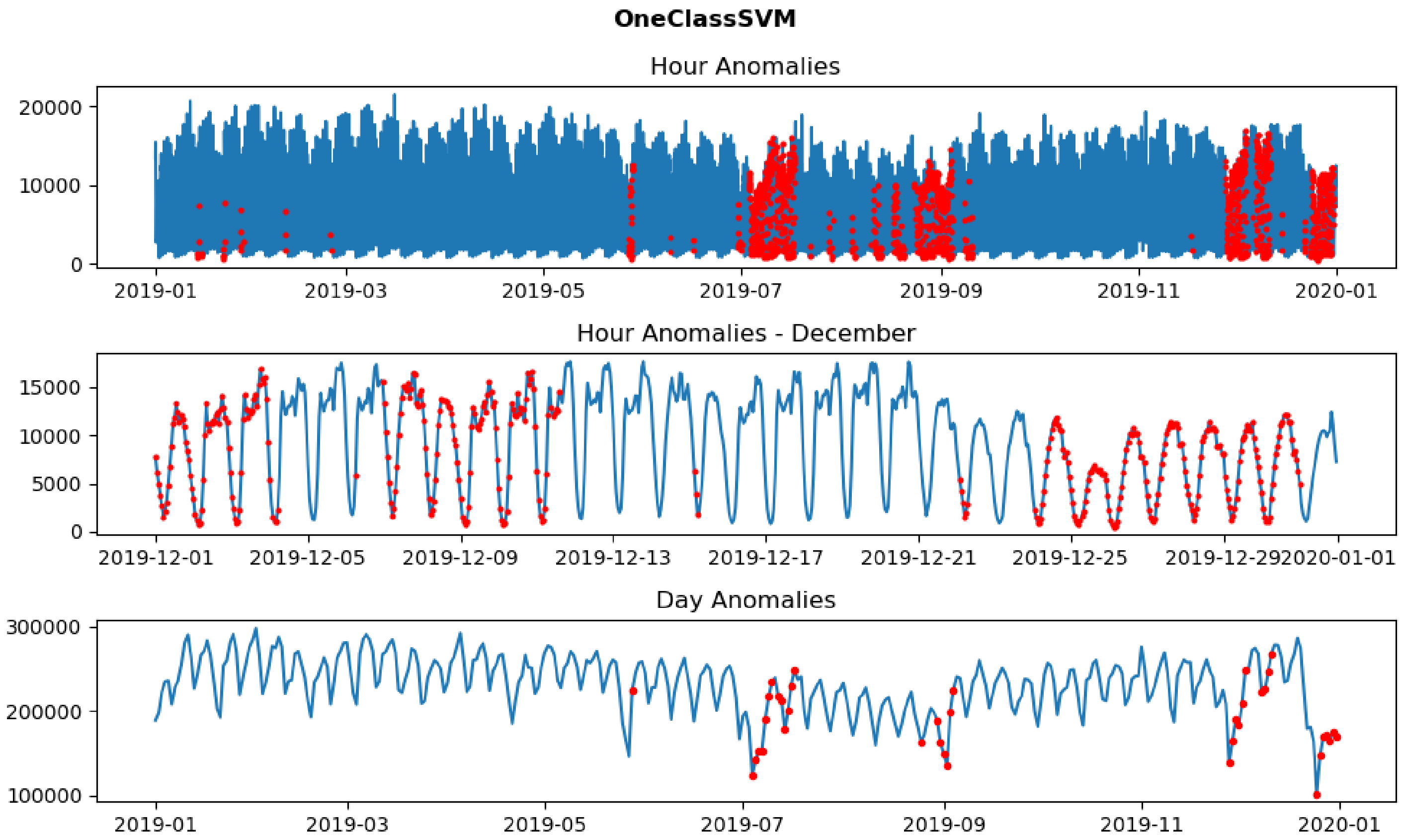


DBSCAN
Create 'normal' clusters while outliers are not clustered.



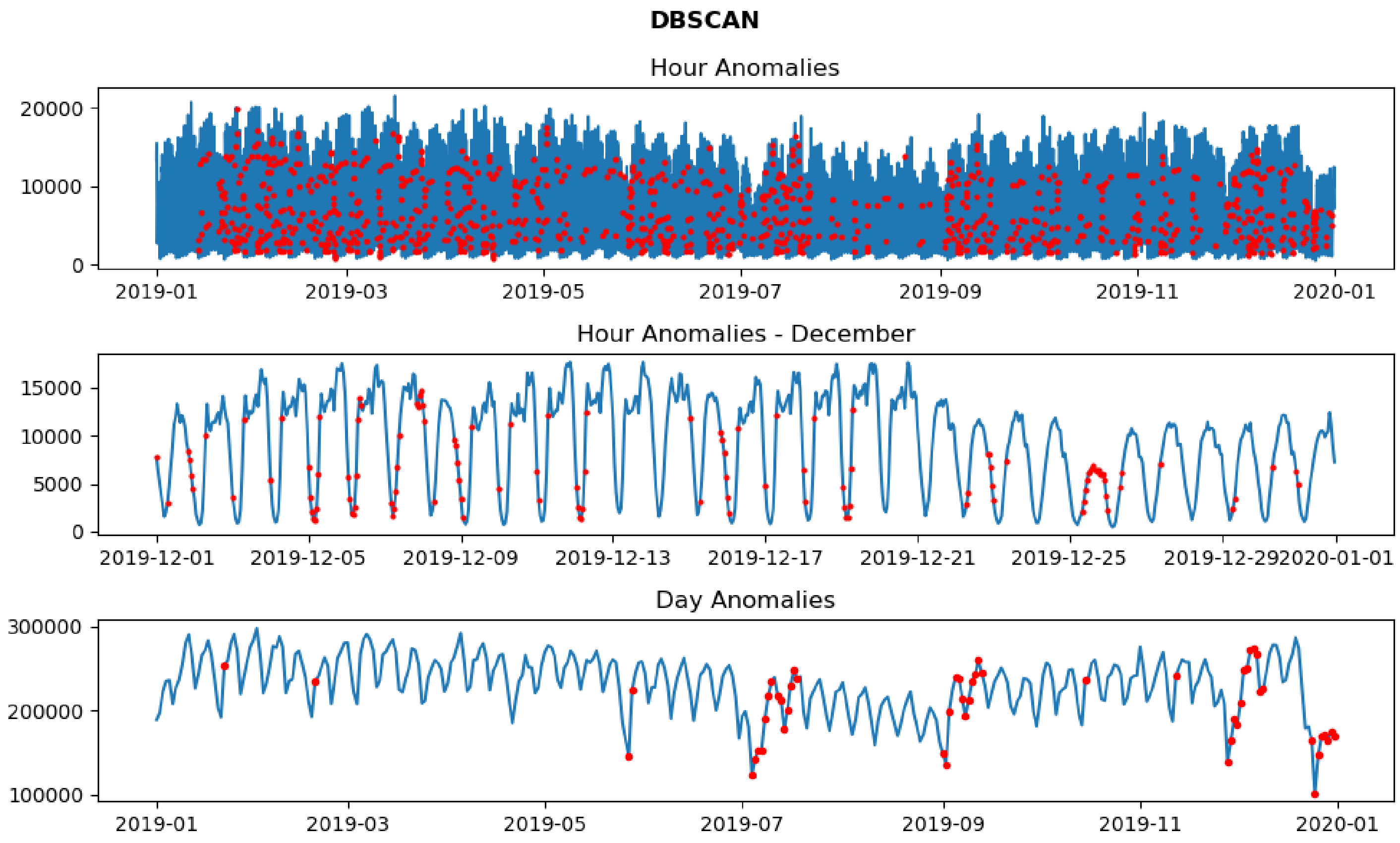
Auto-Encoder
Train on normal data to reduce reconstruction loss. Outliers maintain high reconstruction loss.

Modeling – OneClassSVM



- Hyperparameter tuning on γ (influence of each data point)
  - Reduced γ better generalizes (down to 1E-9!)
  - Adjusted to achieve between 10-15% anomaly
- Clearly marking times of year as anomalies

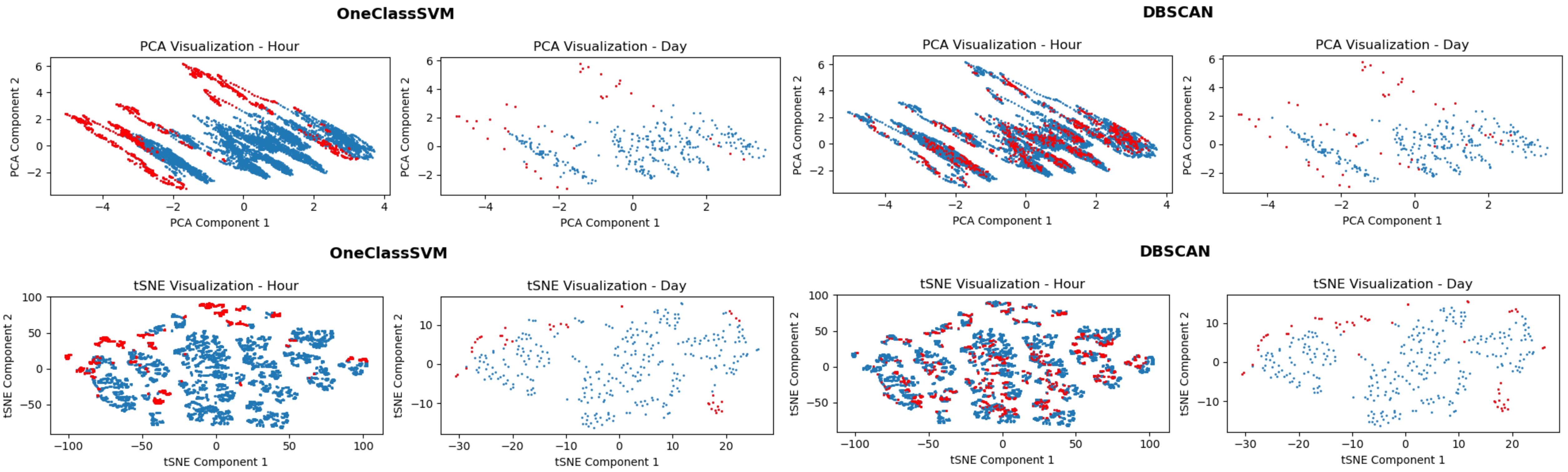
Modeling – DBSCAN



- Hyperparameter tuning on ε (maximum distance between cluster points)
  - Increasing ε creates more outliers
  - Adjusted to achieve between 10-15% anomaly
- Much harder to interpret hourly anomalies

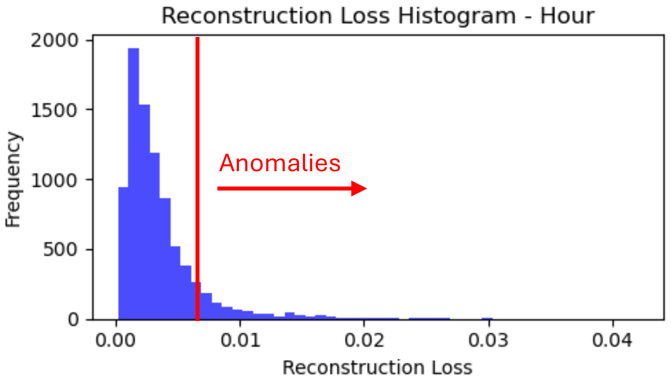
Modeling – OCSVM and DBSCAN

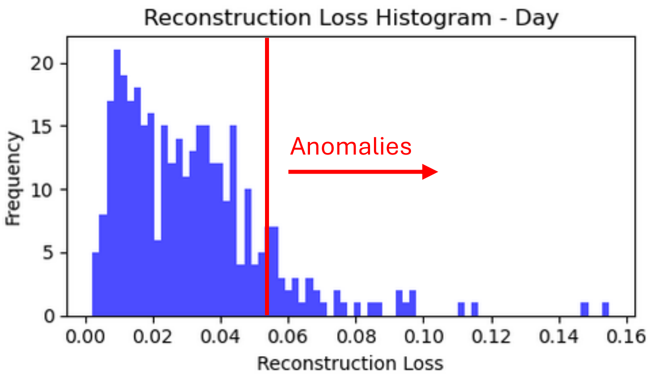
- Differences become apparent comparing PCA and t-SNE

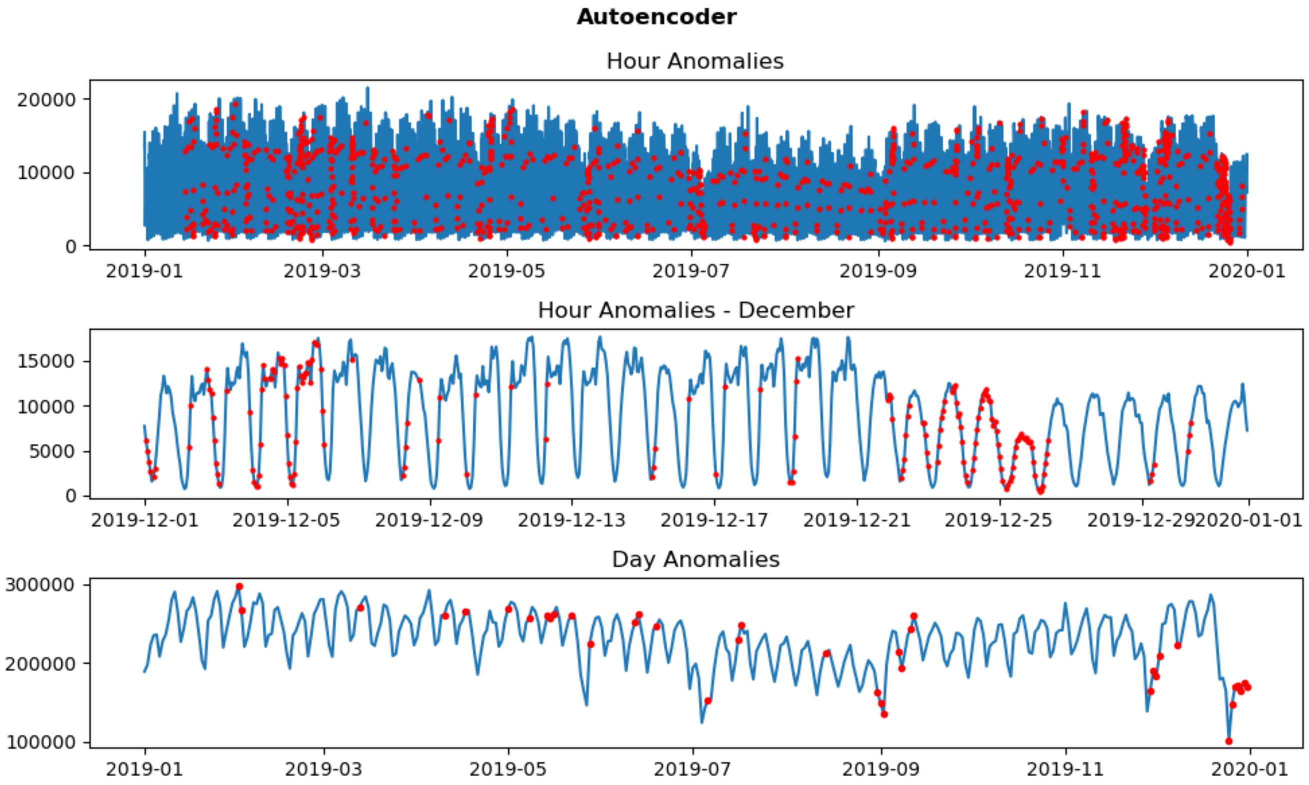


Modeling – Autoencoder

- Train autoencoder to reproduce input where high reconstruction loss indicates anomaly. Seems to have issue with specific hour.

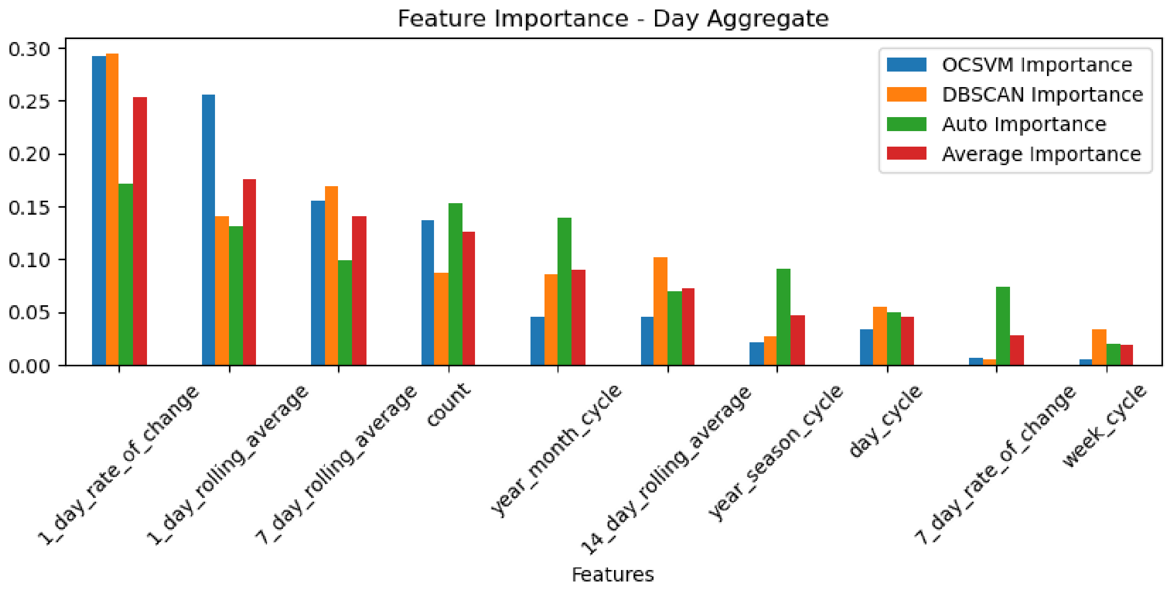


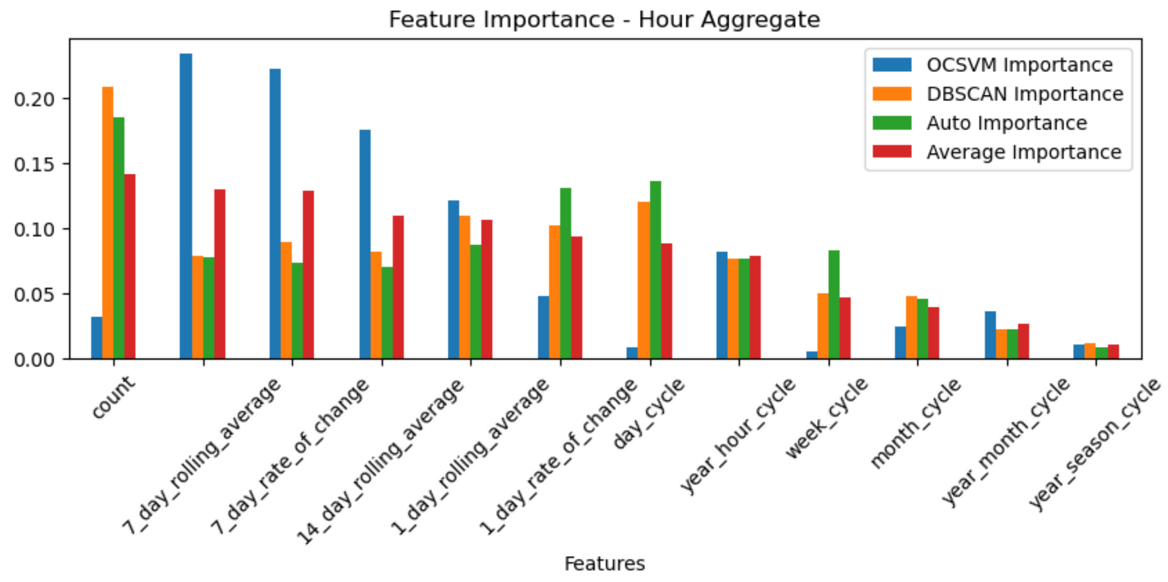




Modeling – Feature Importance

- Supervised Random Forest using each model’s anomaly classifications as labels informs feature importance.
  - While day models appear similar, note significant hour differences.





Evaluation - Analysis

	Model	Hour Outlier %	Day Outlier %
0	OCSVM	70.27	83.33
1	DBSCAM	14.86	91.67
2	Autoencoder	14.86	58.33

- While IQR outliers are not comprehensive, it does reveal differences between the approaches.
- Algorithms performed similarly on simpler Daily-Aggregate data, relying on similar features.
- Wide variation on Hour-Aggregate anomaly labeling, perhaps due to increased hourly variance.
- OCSVM appeared most effective for Hour-Aggregate, effectively grouping into days of anomalies rather than evaluating hours individually.

Discussion

- **Timeline**
  - Project took longer than expected (when does it not).
- **Challenges**
  - A lot of time spent re-working ‘quick-and-dirty’ exploratory code into something report/presentation ready.
  - Original hourly data scope was overwhelming, so added daily aggregate analysis to clearly understand the algorithm performance.
  - Surprisingly, the expected challenges weren’t problematic.
- **Future Mitigations** – What to do better next time
  - Simplify, then simplify again.
  - Spend more time in planning phase to avoid repeated work and to design analysis/functions/graphs from the start.

Conclusion & Future Work

- Anomaly detection in unlabeled time-series is a common issue.
- Algorithms performed similar on day-anomalies, but OCSVM was easier to interpret on harder to parse hour-anomalies.
- Variation likely due to different feature importance, with DBSCAN and autoencoder emphasizing hourly variation.
- PCA and t-SNE visualization reveal other variation in approach.
- **Future Work**
  - Use labeled data, such as Numenta Anomaly Benchmark
  - Test performance on other dataset features.
  - Consider additional feature engineering (holiday, STL decomposition).
  - Compare to LSTM RNN.

Thank you!

Questions, concerns, clarifications

Nick Vastine
DTSA5506 – Data Mining Project
Final Project
April 21, 2024

Data Science
UNIVERSITY OF COLORADO **BOULDER**