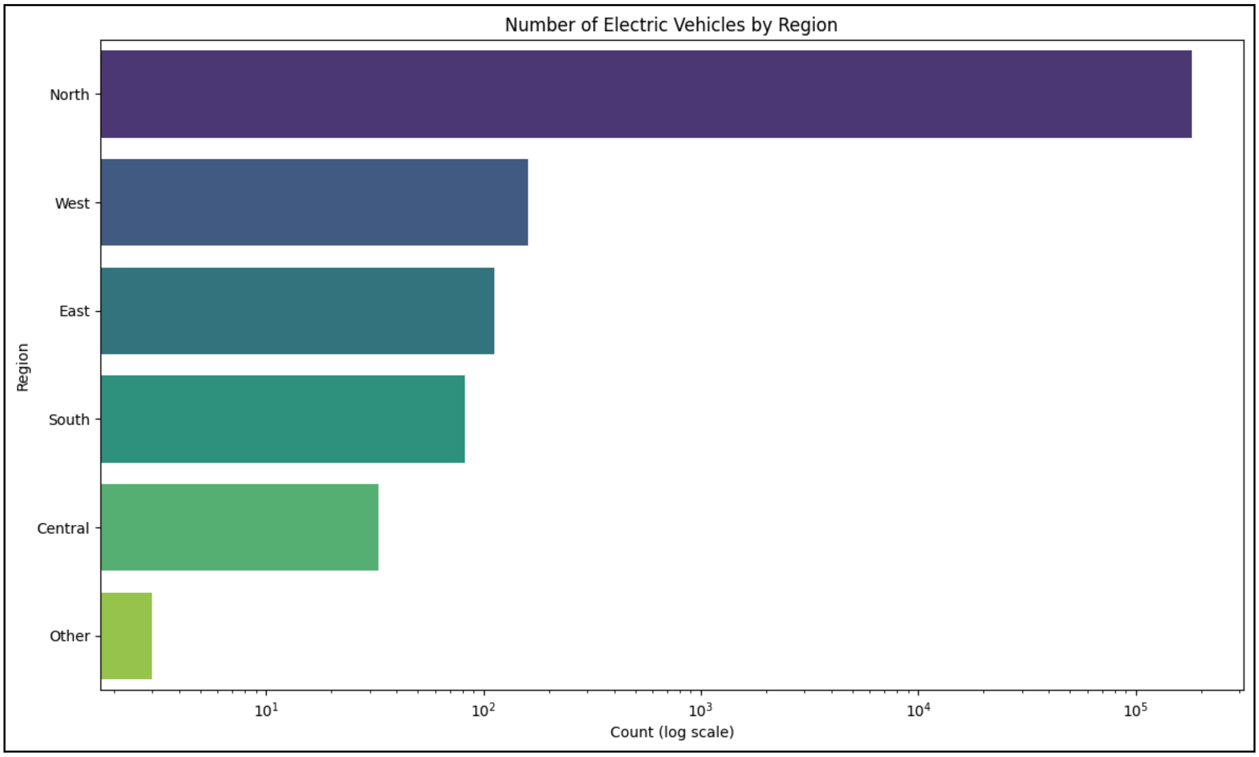
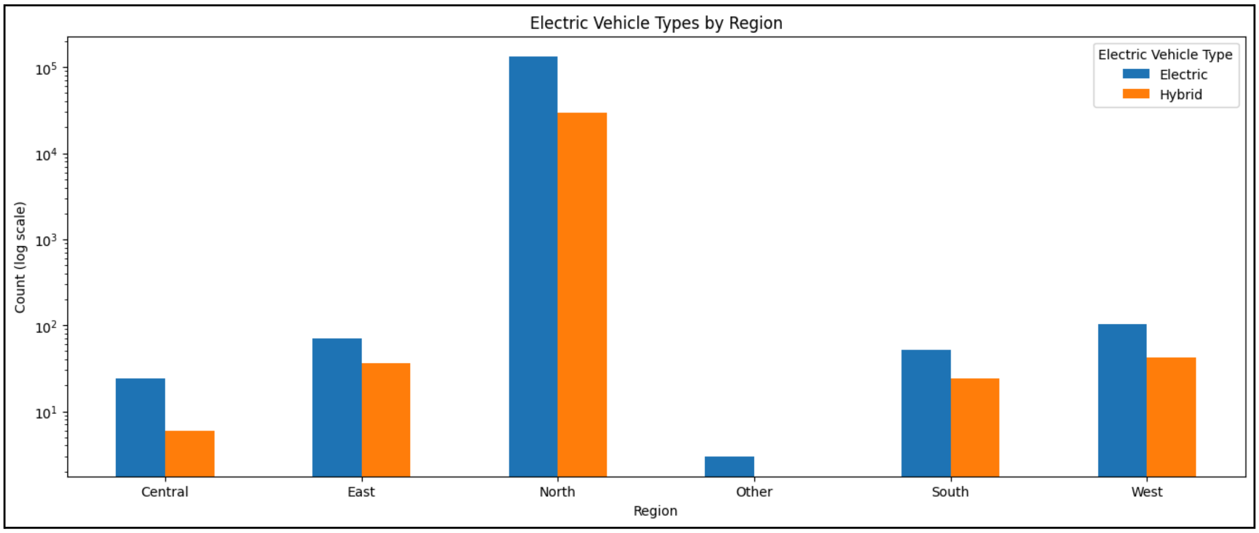
# Assignment 2 - Data Analytics

**Team 31**

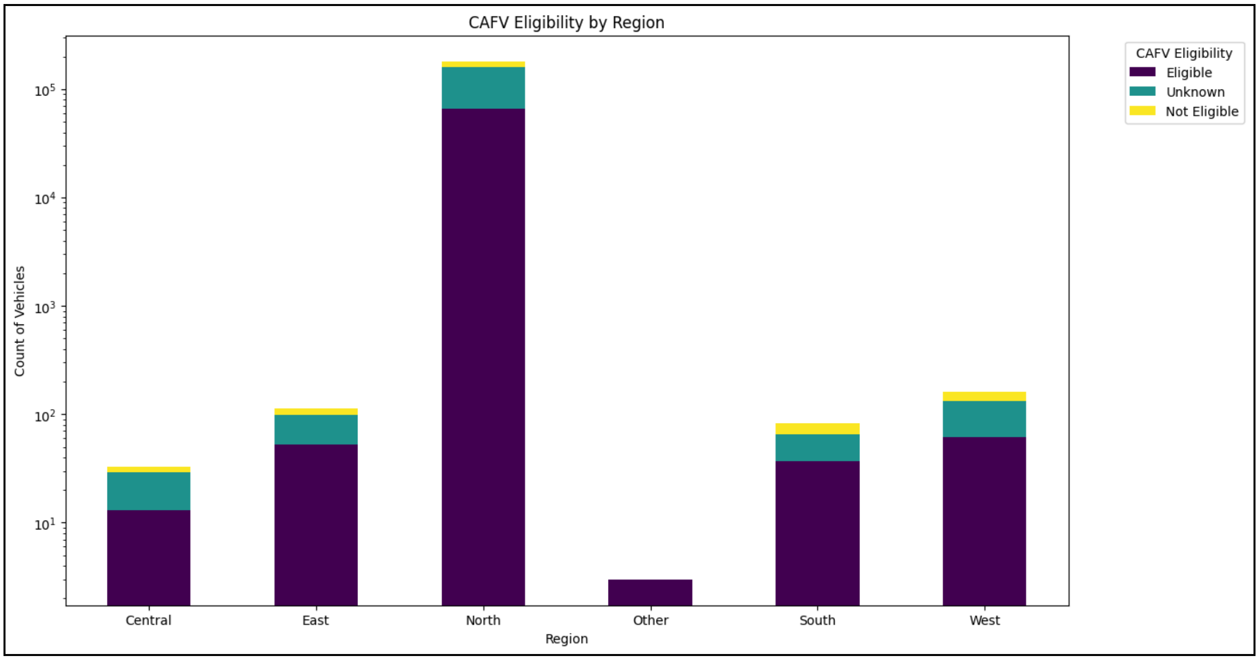
## Attribute-Oriented Induction:



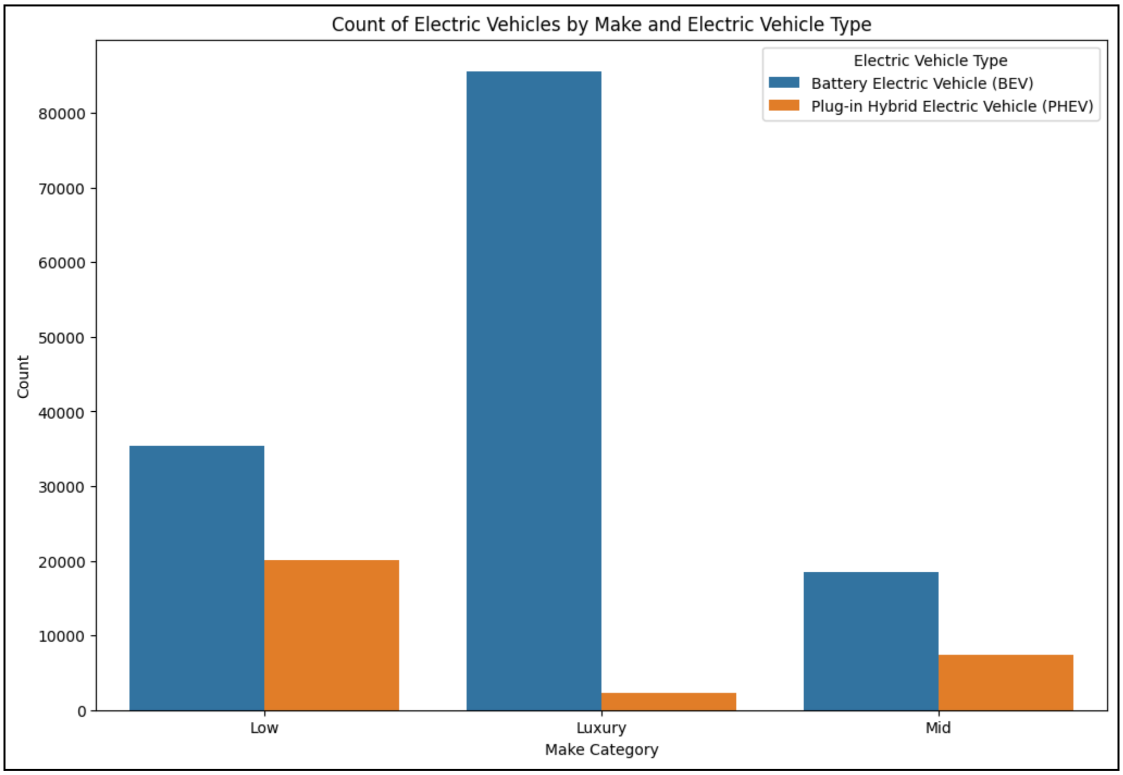
- The North region has the most electric vehicles and Central/ Other(BC, AE) regions have the least.



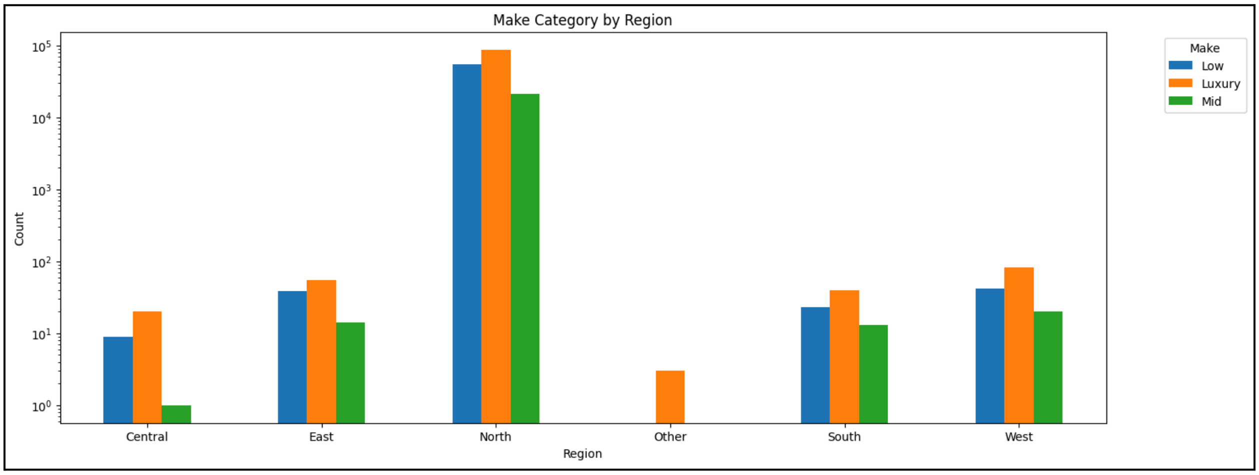
- Central Regions have the most disparity between the 2 electric vehicle types.
- We also infer that Electric cars are preferred over Hybrid in all regions.



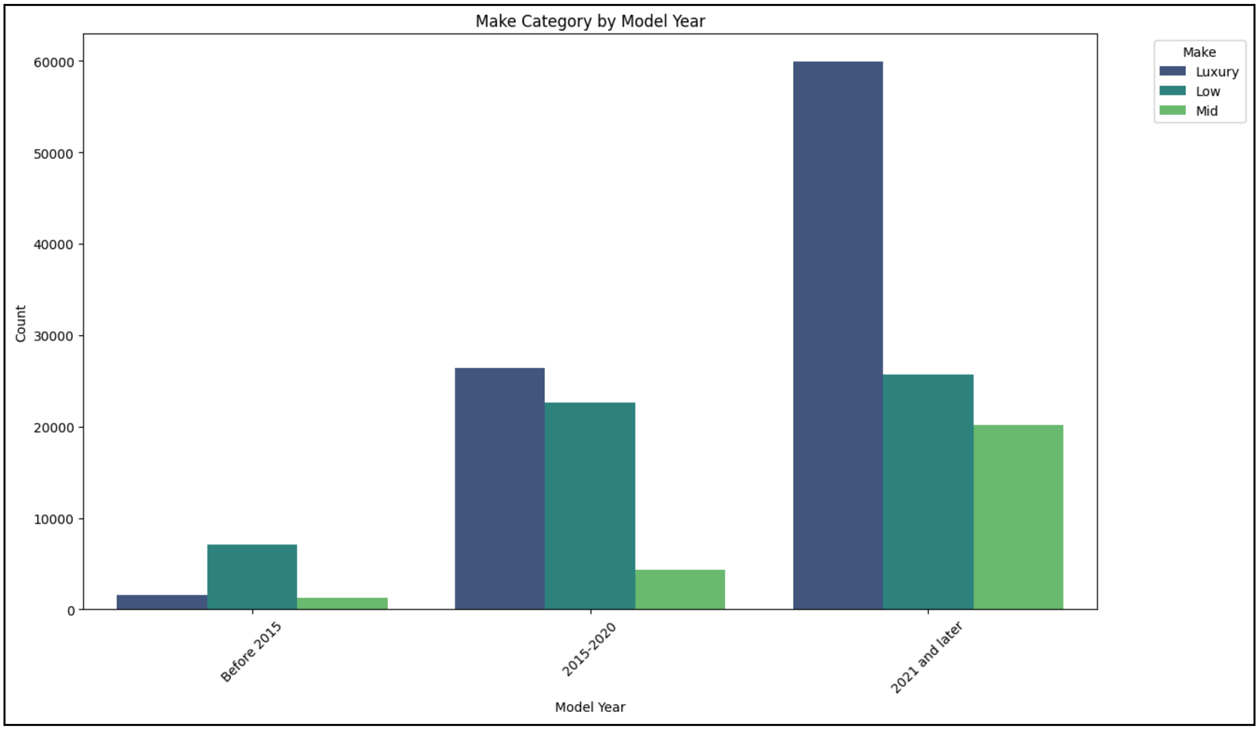
- The North region has the highest number of eligible vehicles.
- Most regions have a small proportion of vehicles with unknown or not eligible status.



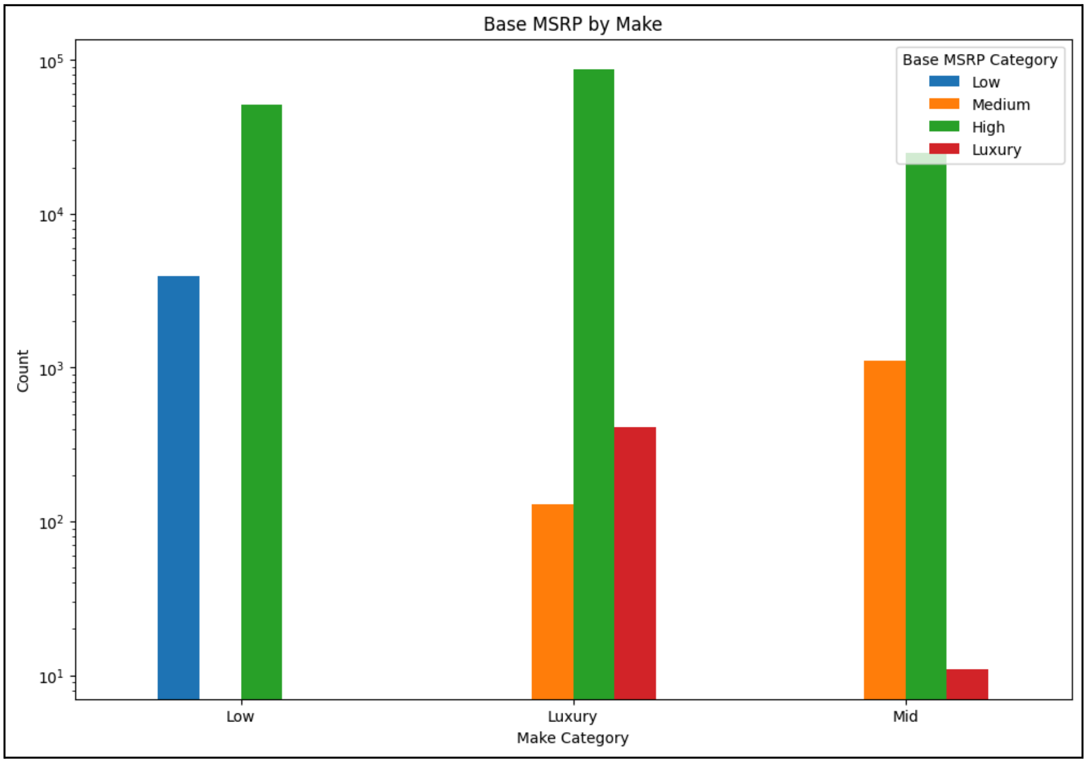
- Luxury vehicles have the highest number of BEVs, significantly more than PHEVs.
- Low and Mid categories have more balanced numbers between BEVs and PHEVs.



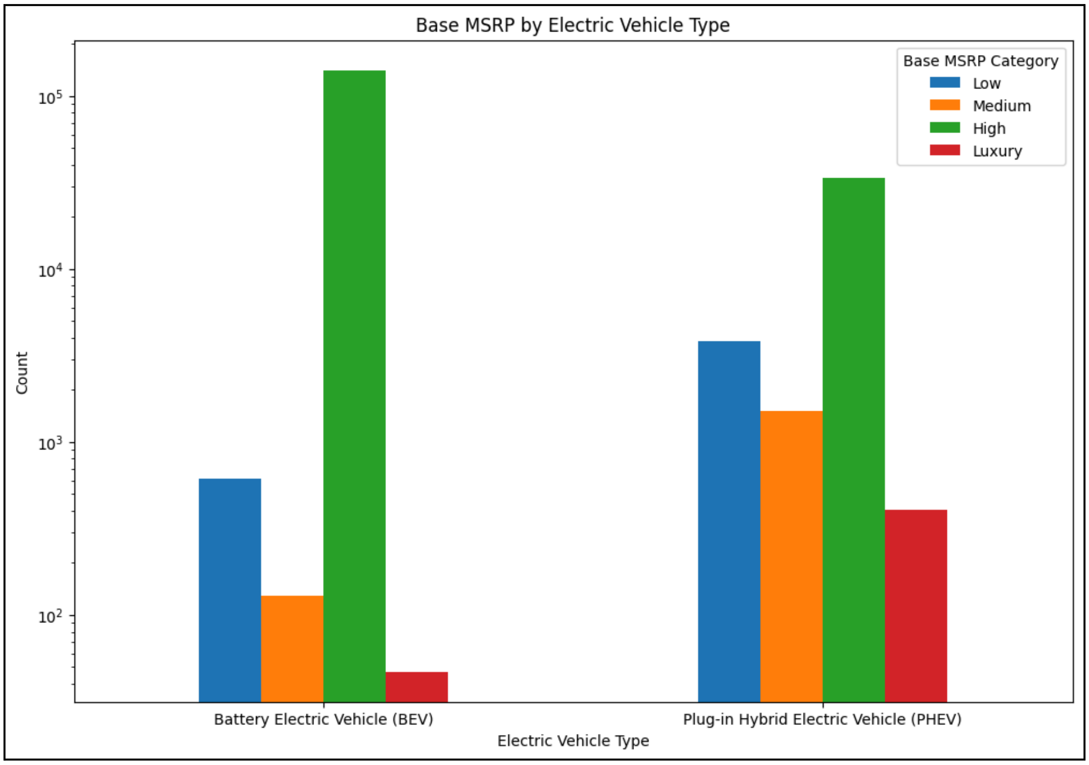
- The North region has the highest number of vehicles in all categories.
- Luxury vehicles are the most common in most regions, followed by Low, then Mid category vehicles.



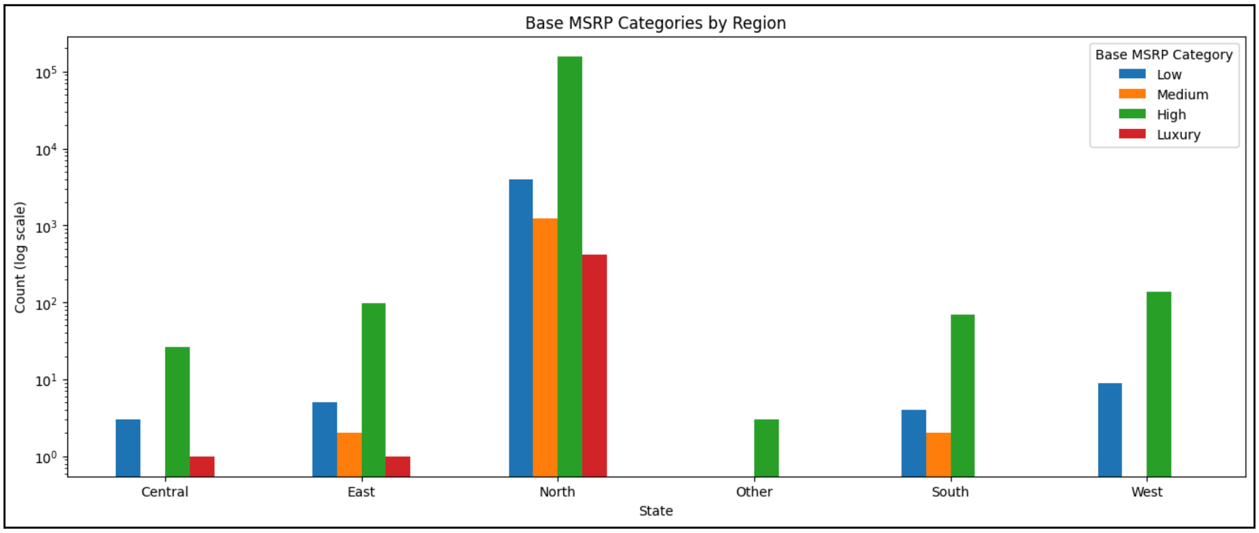
- Luxury vehicles dominate in recent years (2021 and later), showing significant growth over time. Low and Mid categories also increase over time, but less dramatically than Luxury vehicles.
- Cars made before 2015 were generally priced low.
- Cars made in 2015-2020 are mostly luxury ones.



- High MSRP vehicles are most common across all make categories.
- Luxury makes have a more balanced distribution across MSRP categories compared to Low and Mid makes.



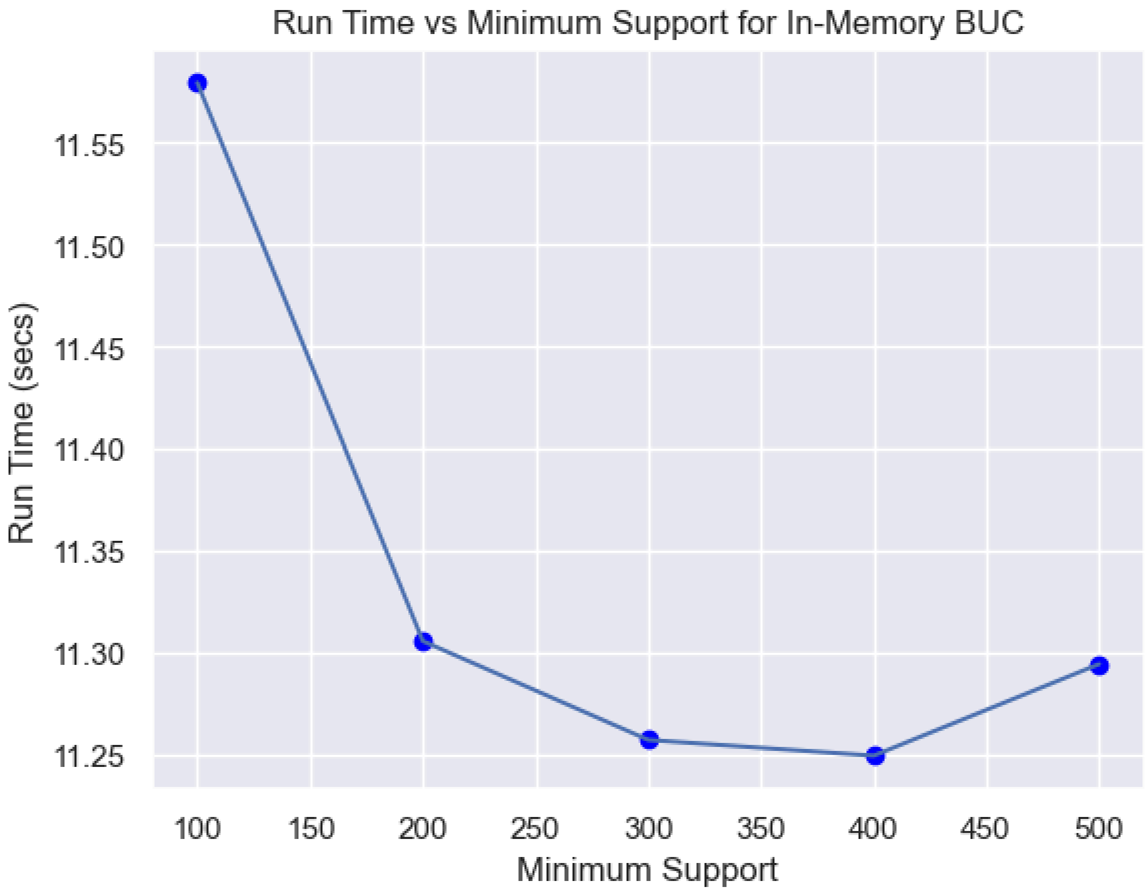
- High MSRP vehicles dominate both BEV and PHEV categories.
- PHEVs have a more even distribution across MSRP categories compared to BEVs.



- North region tend to prefer high-priced cars.
- Cars in AE and BC are also highly priced.
- High MSRP vehicles are the most common in all regions where they're present.
- The West and Central regions show no preference for medium-priced cars.
- Luxury cars are very rare except in North regions.
- The North region has significantly more vehicles across all MSRP categories.

## Performance Analysis:

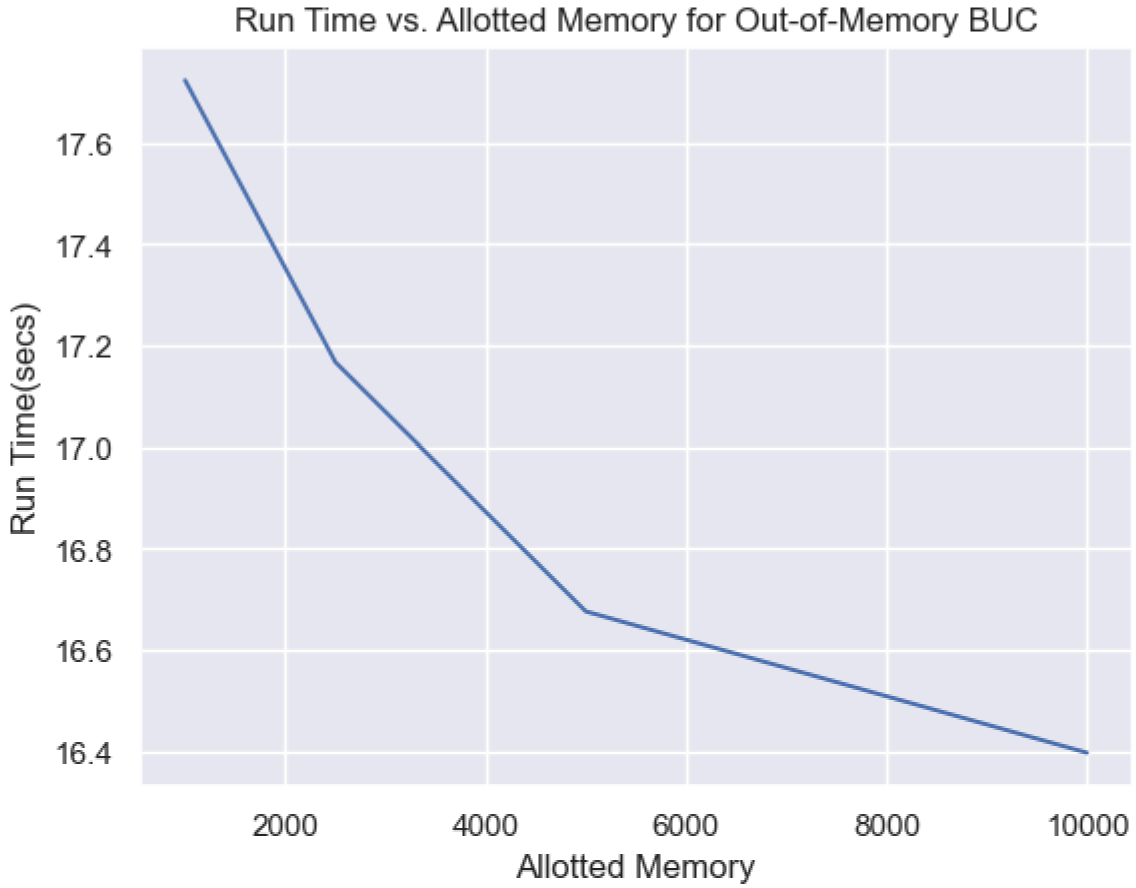
### Run Time vs Minimum Support for In-Memory BUC:



**Inference from the Plot:**

- Displays the relationship between minimum support and run time for in-memory BUC algorithm implementation.
- The decrease in run time is most pronounced between 100 and 200 minimum support. There's a general downward trend in run time as minimum support increases.
- The reason for this is because higher support thresholds lead to fewer qualifying patterns. This leads to a decrease in required computation and hence, runtime.

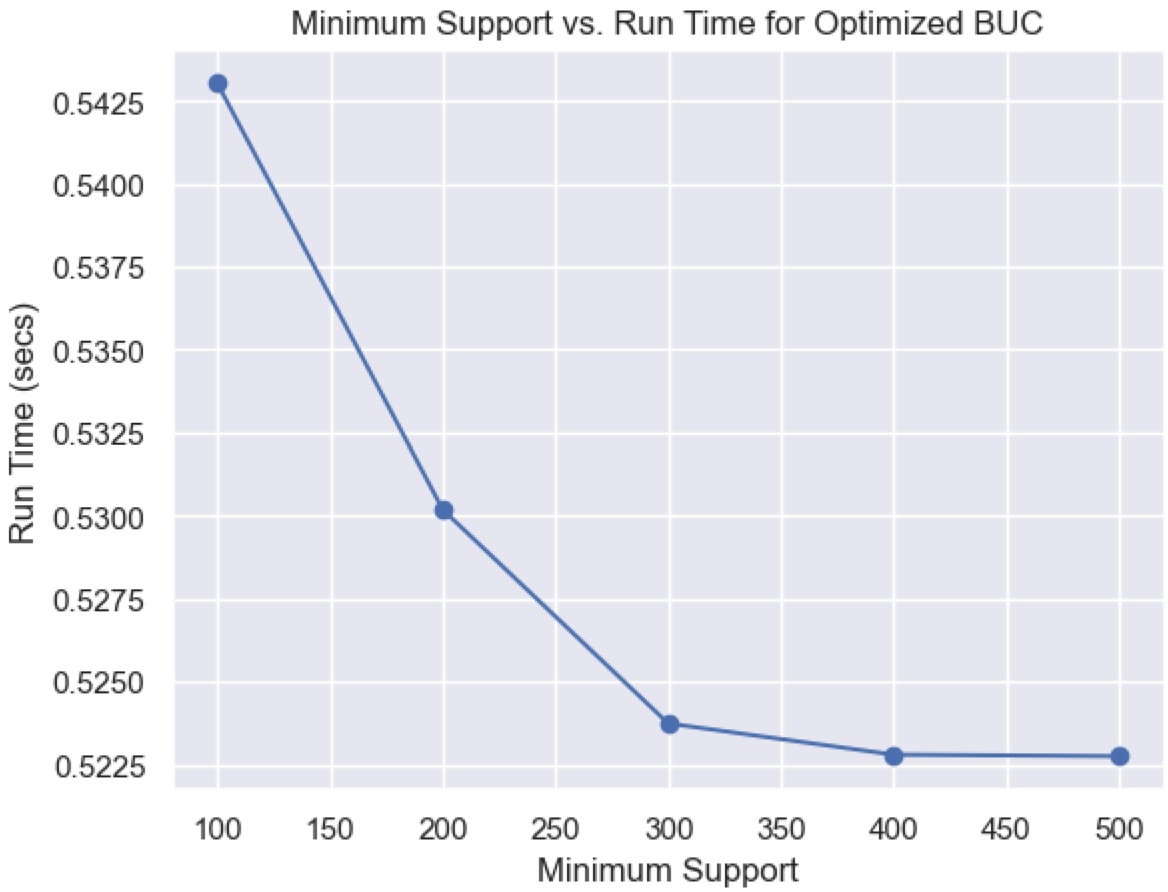
**Runtime vs Allocated Memory for Out-of-Memory BUC:**



**Inference from the Plot:**

- Displays the relationship between allotted memory and run time for out-of-memory BUC algorithm implementation.
- There's a general downward trend in run time as allotted memory increases.
- The reason for this is because fewer pages are required to store data leading to reduction in overhead associated with read/write disk operations which are usually time-consuming.

## Minimum Support vs Run Time for Optimised BUC:



**Inference from the Plot:**

- The graph shows a generally decreasing trend.
- Compared to the original runtime vs min support graph for in-memory BUC, the runtime has significantly decreased for same set of minimum support values. This shows that caching has effectively reduced computation and by extension runtime.

## Comparison of BUC and AOI:

a. **Primary Purposes and Use Cases**

- **BUC (Bottom-Up Cube):**
    - **Purpose**: BUC is used for computing data cubes, which are multidimensional aggregates. The goal is to compute aggregates for all possible combinations of dimensions, allowing for detailed OLAP (Online Analytical Processing) queries.
    - **Use Cases**: It is typically used in data warehousing and business intelligence for summarizing large datasets into group-bys.
- **AOI (Attribute-Oriented Induction):**
    - **Purpose:** AOI is used for conceptually summarizing data by generalizing specific attribute values into higher-level categories. It aims to remove unnecessary detail and identify general patterns.
    - **Use Cases:** AOI is commonly used in data mining for pattern discovery and rule generation. It is effective in domains like market basket analysis, classification, and clustering.

b. **Types of Insights or Patterns**

- **BUC**: Best suited for discovering aggregate patterns over multiple dimensions. Specifically, it is used to compute data cubes. It is focused on quantitative insights (e.g., how many electric vehicles are sold in each state by each manufacturer).
- **AOI**: Best suited for discovering conceptual patterns by reducing the granularity and generalising data. It provides qualitative insights, such as grouping electric vehicles into general price ranges, and generating characteristic rules based on these generalizations.

c. **Computational Efficiency and Scalability**

- **BUC**: Computationally expensive because it computes aggregates for all possible combinations of dimensions (exponentially many combinations). Scalability can be improved with optimizations like Iceberg Cubing, but it can still be slow and memory-intensive for large datasets. The in-memory version is not very scalable due to memory and processing power constraints. The out-memory version is more scalable at the cost of I/O overhead.
- **AOI:** More computationally efficient since it generalizes attributes and reduces the dimensionality of the data. However, the degree of generalization can affect the complexity. It is generally more scalable than BUC because it focuses on summarizing data rather than computing exhaustive aggregates as seen in this implementation.

d. **Interpretability of Results**

   - **BUC:** The results are harder to interpret as they provide detailed numerical summaries (e.g., counts, sums) for all the aggregates and combination of dimensions generated. Due to the large number of combinations, it can be overwhelming and hard to interpret without proper visualization tools.
   - **AOI:** The results are highly interpretable since they summarize the data conceptually, providing clear and concise generalizations. For example, instead of showing the exact MSRP for each vehicle, it groups them into "Low", "Mid", "High" and “Luxury” categories, making it easier to understand broad trends.

e. **Scenarios for Preference**

   - **BUC:**
     - BUC is preferable when you need detailed quantitative analysis across multiple dimensions. It can provide granular insights like analyzing how many electric vehicles of a specific make and model were sold in different states in a given year.
   - **AOI:**
     - AOI is preferable when you want to perform qualitative data mining and generalization. It is useful for discovering higher-level patterns or when the dataset is too granular, and you want to abstract it into broader categories like generalizing electric vehicle sales into broad categories based on the vehicle type, region, and price range.

## Example from Implementations

- Using the BUC algorithm, user can observe how number of electric vehicles change as state and vehicle types vary.
- In AOI, if you wanted to see broad trends in electric vehicle sales, you could generalize the Make and Model into categories (e.g., Luxury, Mid, Economy), and then identify high-level patterns such as “Most luxury electric vehicles are sold in North of America.”

## Conclusion:

- BUC is powerful for detailed, multidimensional analysis but can be computationally expensive and harder to interpret without aggregation tools.
- AOI is more scalable and offers clearer high-level insights, making it ideal for pattern discovery and conceptual summarization.