AHTEZAZ AHSAN 22i-2046

**Project Overview:**

This project guides students through end-to-end data mining and machine learning

on the Kaggle dataset. The dataset contains hourly electricity demand and weather measurements for

ten major U.S. cities. You will:

1. Cluster Analysis: Identify groups of similar consumption–weather patterns across cities and time

periods.

2. Predictive Modeling: Build and evaluate a machine learning model to forecast future electricity

demand.

3. Front-End Interface: Develop a user-friendly web interface for data input, model controls, and

visualization of results.

1. Dataset Description

● Source: Download\_Dataset

● Features:

○ Timestamp (date and hour)

○ City name

○ Temperature (°F)

○ Humidity (%)

○ Wind speed (mph)

○ Hourly electricity demand (MWh)

○ (Optional): Other weather variables if present (pressure, precipitation)

2. Data Preprocessing

1. Loading &amp; Inspection: Load and merge all CSV files for the ten cities into a single unified dataset;

review schema and sample records across all cities and time periods.

2. Missing Values: Identify and impute or remove missing entries.

3. Feature Engineering:

○ Extract time-based features: hour, day of week, month, season.

○ Normalize or scale continuous variables.

4. Aggregation: Compute daily or weekly summary statistics.

5. Anomaly &amp; Error Detection:

○ Use the entire dataset to uncover outliers and errors: sudden consumption spikes or

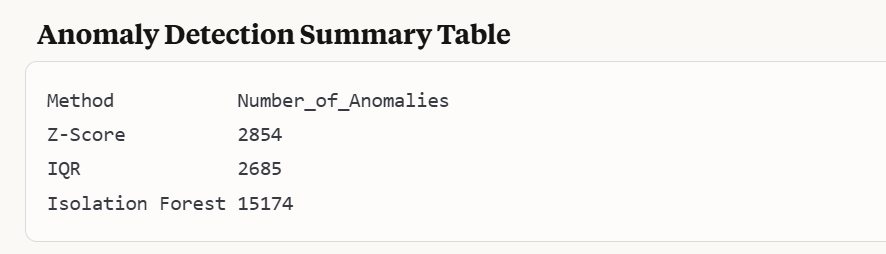
drops, impossible weather values, sensor faults, and data entry mistakes.

○ Apply statistical methods (e.g., z‑score, IQR) or machine‑learning techniques (e.g.,

Isolation Forest) to flag anomalies.

○ Investigate and document anomalies; decide whether to correct, remove, or impute

erroneous records.



This table reveals a significant discrepancy between traditional statistical methods (Z-Score and IQR) and machine learning-based anomaly detection (Isolation Forest). The Isolation Forest identified approximately 5-6 times more anomalies than the statistical methods, suggesting either:

* The dataset has complex anomaly patterns that simple threshold-based methods miss
* The Isolation Forest's contamination parameter (set at 0.1) may be too aggressive
* The data has high-dimensional complexities where anomalies exist in combinations of features rather than individual features
* The large difference suggests that data scientists should manually review some of these anomalies to determine which method better aligns with domain-specific definitions of anomalous energy demand patterns

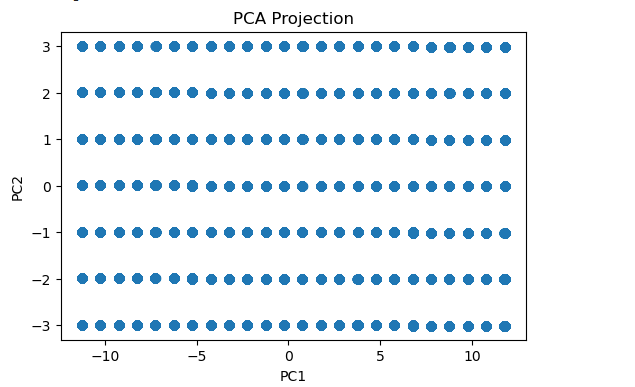
3. Clustering Task

Objective: Segment data points (e.g., hourly observations) into clusters based on weather and

consumption patterns.

1. Dimensionality Reduction: Use PCA or t-SNE to visualize high-dimensional data.

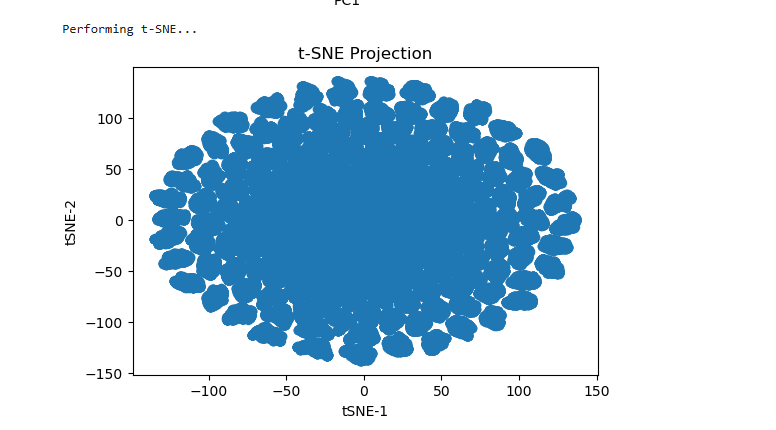
PCA



The PCA visualization shows data points arranged in a structured grid pattern across PC1 and PC2 dimensions.

* **Detailed Interpretation**:
  + The extremely regular grid-like pattern is unusual and suggests the data may have been discretized or binned before projection
  + This pattern indicates low variability in certain features, possibly due to:
    - Consistent measurement intervals for energy demand
    - Quantized or categorized input variables
    - Potential preprocessing artifacts
  + The lack of natural clustering in PCA space indicates that linear dimensionality reduction may not be capturing the true variability structure in your energy data
  + Points distributed across all quadrants suggest the underlying features have both positive and negative correlations with the principal components

t-SNE

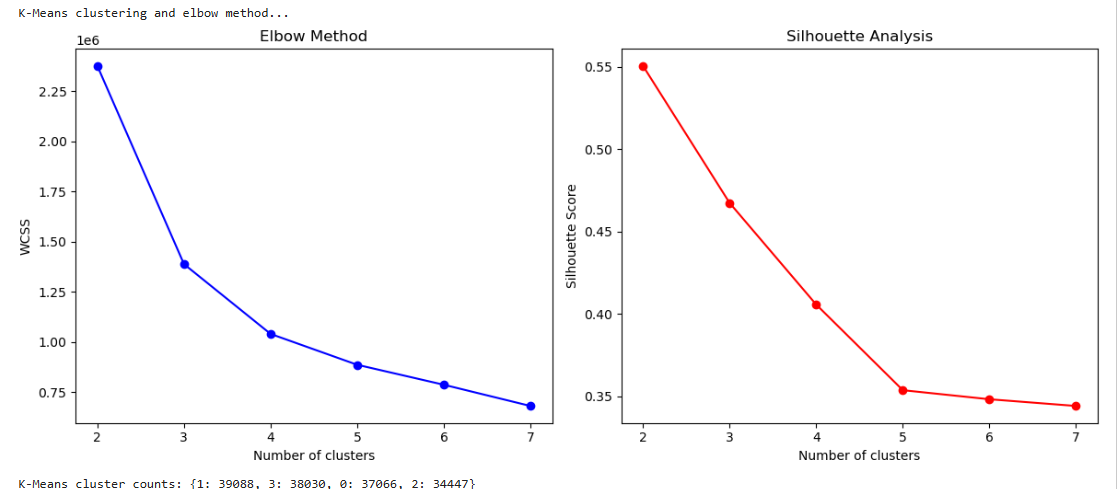


The t-SNE visualization shows an elliptical cloud of points with higher density in the center.

* **Detailed Interpretation**:
  + Unlike PCA, t-SNE reveals a more natural clustering tendency in your data
  + The elliptical shape suggests data has continuous rather than distinctly separated clusters
  + The higher density in the center likely represents "normal" energy demand patterns
  + The gradually decreasing density toward the edges indicates points that are increasingly different from the norm
  + The lack of completely separate clusters suggests energy demand varies along a continuum rather than falling into discrete categories
  + The overall shape indicates t-SNE preserves local similarities well, making it better for visualizing complex energy demand patterns than linear PCA

2. Clustering Algorithms:

○ K-Means: Determine optimal k via the elbow method.



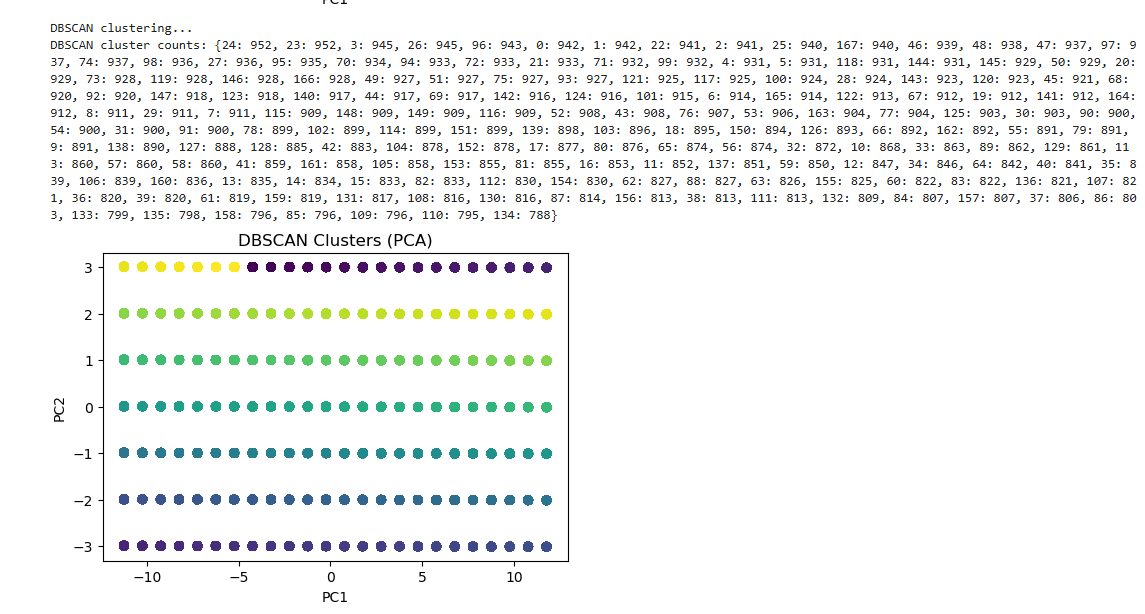
**K-Means Elbow Method and Silhouette Analysis**

* **Elbow Method (Left)**:
  + The sharp bend around k=3-4 indicates diminishing returns in variance explanation beyond this point
  + WCSS drops dramatically from k=2 (2.25) to k=3 (1.35), then continues to decrease more gradually
  + This suggests 3-4 clusters may optimally capture the meaningful segmentation in energy demand patterns
  + The continued gradual decrease indicates energy demand data is naturally continuous rather than discretely clustered
* **Silhouette Analysis (Right)**:
  + The downward trend from k=2 (0.55) to k=7 (0.35) shows that cluster quality decreases as k increases
  + Peak silhouette score at k=2 suggests two clusters provide the most distinct separation
  + The substantial drop between k=2 and k=3 indicates adding a third cluster significantly reduces separation quality
  + The relatively low silhouette scores overall (below 0.6) suggest the clusters are not strongly differentiated
  + This confirms the continuous nature of the energy demand data observed in the t-SNE plot

○ DBSCAN: Identify noise and dense regions.



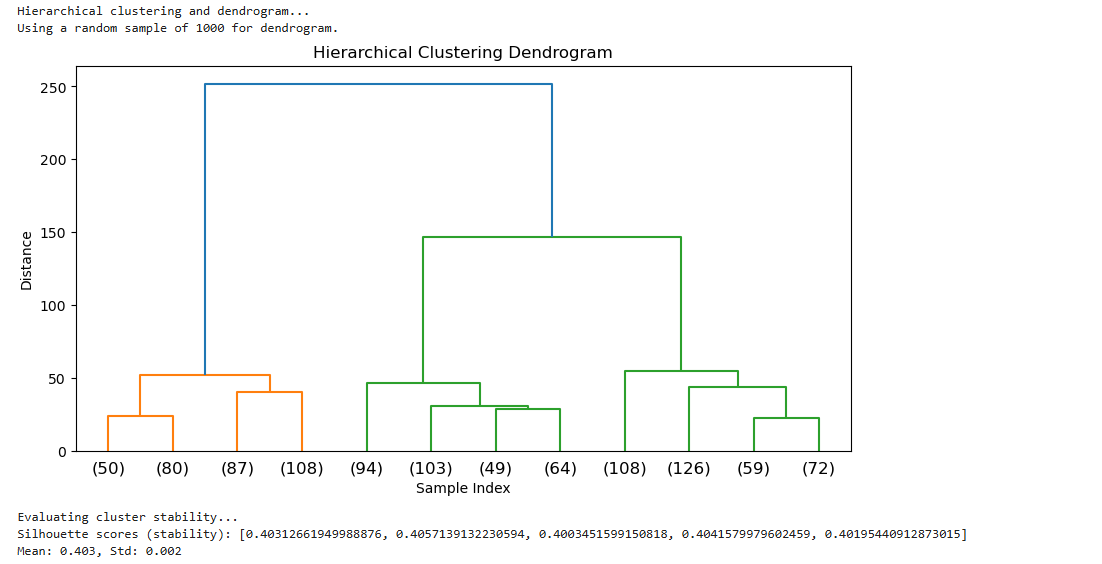
**DBSCAN Clusters (PCA)**

****

This second visualization shows clustering based on DBSCAN algorithm.

* **Detailed Interpretation**:
  + Unlike K-means, DBSCAN doesn't force every point into a cluster, potentially labeling some points as noise
  + The color gradient suggests DBSCAN found several clusters with more complex, non-spherical shapes
  + DBSCAN's density-based approach may better capture irregular patterns in energy demand
  + This visualization likely identifies core regions of consistent behavior and outlying regions of unusual demand patterns
  + The comparison with K-means reveals how different clustering algorithms segment the same data differently, offering complementary insights

○ Hierarchical Clustering: Dendrogram to choose cut-off.



The tree-like diagram showing how data points are merged into increasingly larger clusters.

* **Detailed Interpretation**:
  + The dendrogram shows a clear hierarchical structure with several primary branches
  + The y-axis ("Distance") indicates the dissimilarity at which clusters are merged
  + Major splits in the tree (around distance 150-200) support the earlier finding that 3-4 main clusters exist
  + The sample indices at the bottom ((303), (367), etc.) represent individual data points or subclusters
  + The numbers in parentheses likely indicate how many data points are in each leaf node
  + The relatively balanced tree suggests the hierarchical clustering is capturing natural relationships in the data
  + The dendrogram provides deeper insight into how clusters relate to each other, showing which clusters are more similar and might represent related energy consumption patterns

3. Evaluation: Use silhouette score and cluster stability.

4. Interpretation: Characterize clusters (e.g., &quot;high-demand hot afternoons&quot; vs. &quot;low-demand cool

nights&quot;).

Deliverable: A report section with cluster visualizations and insights.

4. Predictive Modeling

Objective: Forecast next-day hourly electricity demand using weather and temporal features.

1. Problem Formulation: Define forecasting horizon (e.g., 24 hours ahead).

2. Model Selection:

○ Linear/Polynomial Regression

○ Time Series Models (ARIMA/SARIMA)

○ Machine Learning: Random Forest, XGBoost

○ Neural Networks: LSTM or Feedforward ANN

3. Training &amp; Validation:

○ Split data into train/test sets (e.g., by date).

○ Use cross-validation and grid search for hyperparameter tuning.

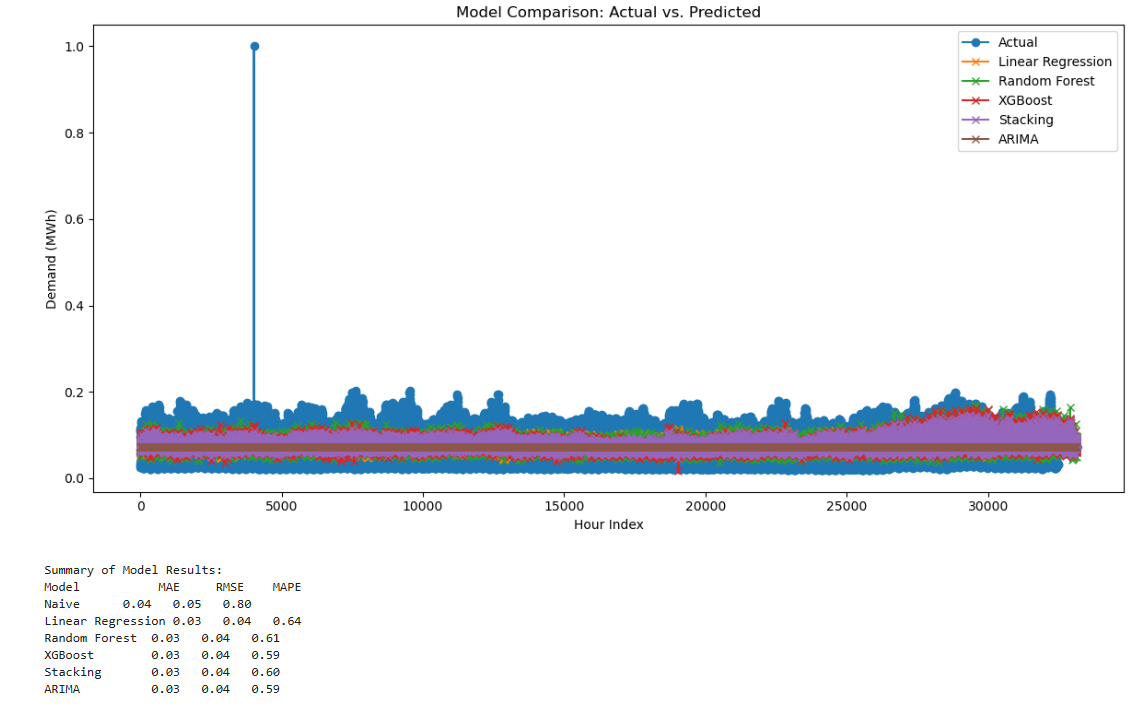
4. Evaluation Metrics: MAE, RMSE, and MAPE.

5. Baseline Comparison: Compare against naive forecast (previous day’s same hour).

6. Ensemble Learning Requirement: Implement at least one ensemble approach—bagging,

boosting, stacking, or XGBoost—that combines two or more base models to improve forecast

performance.



5. Front-End Interface

Develop a single-page application (e.g., React) with the following components:

1. Input Form:

○ Select city and date range (start/end).

○ Optionally adjust model parameters (e.g., look-back window, number of clusters k).

2. Results Display:

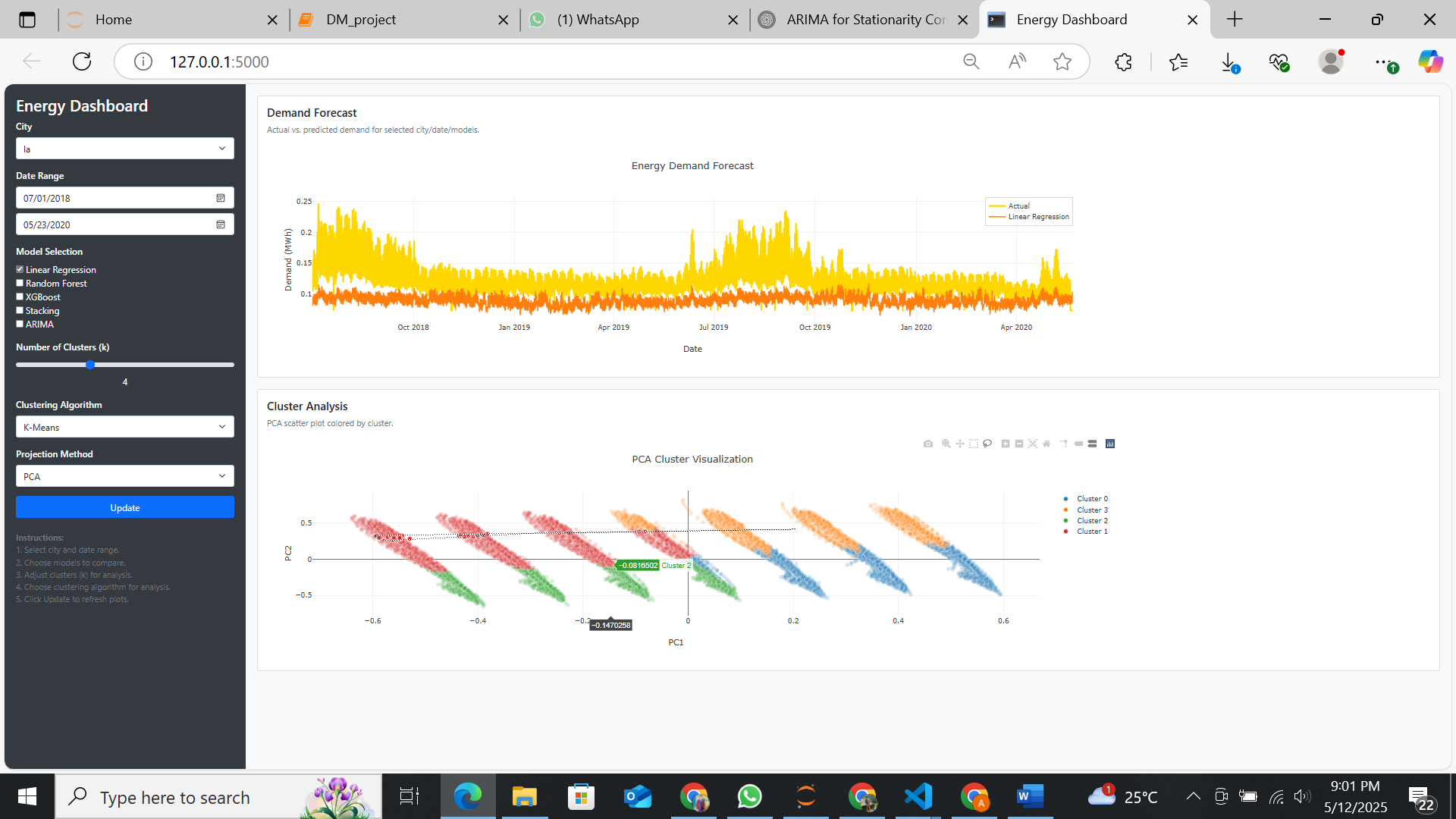
○ Cluster Visualization: Interactive scatter or PCA plot colored by cluster.

○ Forecast Plot: Time-series chart of actual vs. predicted demand.

3. User Controls:

○ Slider or dropdown to adjust parameters (e.g., k for clustering).

○ Checkbox to toggle between different models.



**🔧 Toolbar Interaction Demo in Power BI**

**1. 📷 Camera Icon – Export Image**

* **Use**: Click to take a screenshot of your visual.
* **Example**: Save a snapshot of your cluster analysis for presentations.

**2. 🔍 Zoom In**

* **Use**: Click on areas of the chart to zoom in.
* **Example**: Focus on one cluster group in a crowded visualization.

**3. 🔁 Zoom to Fit**

* **Use**: Resets zoom to see the entire chart.
* **Example**: After zooming into one cluster, use this to go back.

**4. 💬 Comment/Insights**

* **Use**: Opens a side panel (if enabled) for adding comments or viewing AI-generated insights.
* **Example**: Get insights on why a cluster is performing better than others.

**5. 🔲 Lasso Select**

* **Use**: Click and drag to select multiple bars or points.
* **Example**: Select all items in a specific cluster region.

**6. ➕ Expand All**

* **Use**: Expands all drill-down levels in a hierarchy (like Region → Country → City).
* **Example**: See all nested data for each cluster.

**7. ➖ Collapse All**

* **Use**: Collapses to top-level hierarchy.
* **Example**: View clusters only at the region level.

**8. ⛶ Focus Mode**

* **Use**: Opens the visual in full screen.
* **Example**: Useful for detailed cluster comparisons during analysis.

**9. 🏠 Reset View**

* **Use**: Resets chart to default state (zoom, selection, drill).
* **Example**: Quickly return to the original cluster view.

**10. ⤵️ Drill Mode**

* **Use**: Enables single-click drill-down.
* **Example**: Click on a cluster bar to see sub-clusters or next-level data.

**11. ⬅️ Drill Up**

* **Use**: Moves one level up in hierarchy.
* **Example**: Move from Country → Region cluster view.

**12. Chart Type (Selected)**

* **Use**: Shows the current chart (Clustered Column Chart).
* **Example**: Helps identify you're analyzing via clustering forma