**Electric Load Forecasting Using Data Mining Techniques**

**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](https://www.kaggle.com/datasets/shubhamkulkarni01/us-top-10-cities-electricity-and-weather-data)
* **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 & 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 & 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.

**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.
2. **Clustering Algorithms:**
   * K-Means: Determine optimal k via the elbow method.
   * DBSCAN: Identify noise and dense regions.
   * Hierarchical Clustering: Dendrogram to choose cut-off.
3. **Evaluation:** Use silhouette score and cluster stability.
4. **Interpretation:** Characterize clusters (e.g., "high-demand hot afternoons" vs. "low-demand cool nights").

**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 & 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.

Deliverable: A Jupyter notebook with code, visualizations, and performance summary.

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

Deliverable: A Jupyter notebook with code, visualizations, and performance summary.

## **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.
4. **Help & Documentation:**
   * Instructions on using each control.
   * Explanation of clustering and forecasting approach.
   * Technical details: data sources, algorithms, and metrics.

**Submission Requirements**

* **Jupyter Notebooks:** Data preprocessing, clustering, and forecasting code.
* **Front-End Code:** Source files for the web interface.
* **Report:** PDF or Markdown summarizing methods, results, and discussion on what you did and how you did it.