Electricity Demand Forecasting System: A Machine Learning Approach

TEAM:

Names: Aaqib Ahmed Nazir, Ammar Kashif
Roll No.: i22-1920, 22i-1968
Section: DS-D

Abstract:

This report details the development and implementation of a machine learning-powered system for electricity demand forecasting. The system provides accurate demand predictions, performs clustering analysis of usage patterns, and offers an interactive AI assistant for energy-related inquiries. By leveraging historical consumption data, weather conditions, and temporal factors, the system delivers valuable insights for energy providers to optimize resource allocation, enhance grid stability, and support the integration of renewable energy sources. The application utilizes a robust combination of technologies, including Python, FastAPI, React, and advanced machine learning models such as XGBoost and LSTM networks.

1. Introduction

Accurate electricity demand forecasting is crucial in the energy sector for several reasons. It enables effective grid management, optimizes resource allocation, facilitates the integration of renewable energy sources, and supports economic planning. This project aims to develop a system that provides precise electricity demand forecasts, identifies usage patterns, and offers an intuitive platform for users to understand and analyze energy consumption data.

1.1 Importance of Electricity Demand Forecasting

The significance of accurate electricity demand forecasting can be summarized as follows:

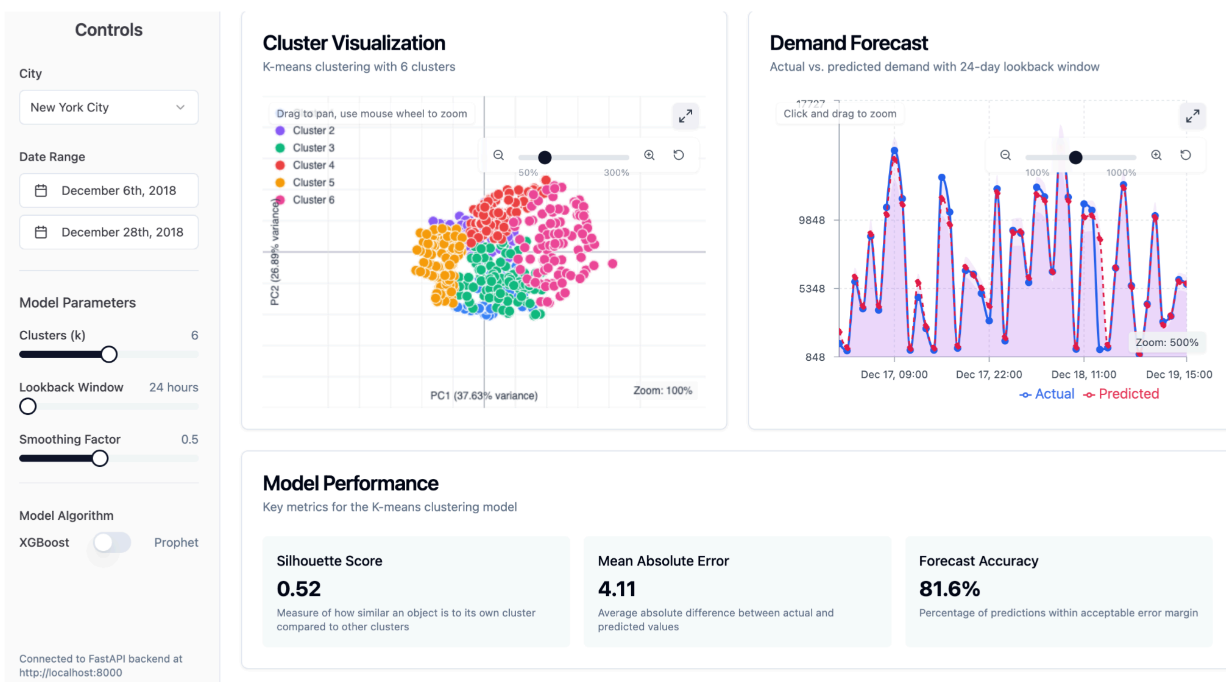
- **Grid Stability:** Balancing electricity supply and demand is essential for maintaining grid stability and preventing blackouts or brownouts.
- **Resource Optimization:** Accurate forecasts enable energy providers to optimize the generation, transmission, and distribution of electricity, reducing waste and operational costs.
- **Renewable Integration:** Forecasting plays a vital role in managing the intermittent nature of renewable energy sources like solar and wind power, ensuring a reliable energy supply.
- **Environmental Impact:** Efficient energy use, guided by accurate demand forecasts, helps minimize the carbon footprint of electricity generation.
- **Economic Planning:** Reliable forecasts support informed decision-making regarding investments in energy infrastructure and market operations.

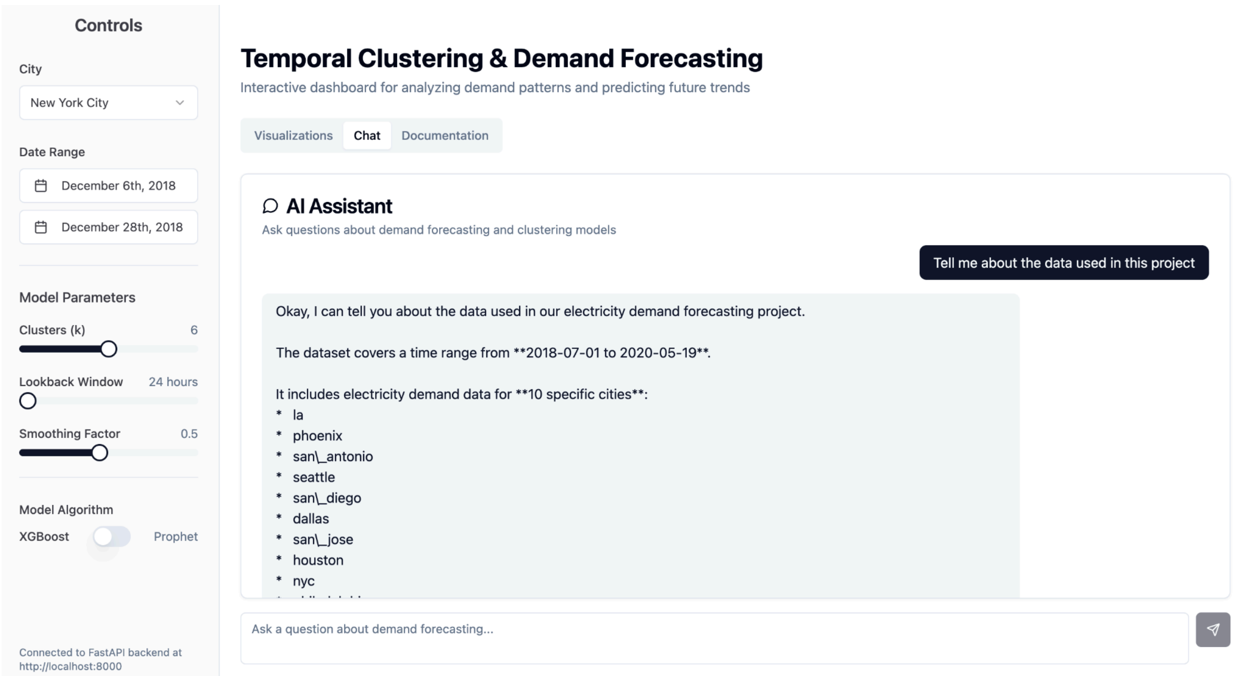
2. System Overview

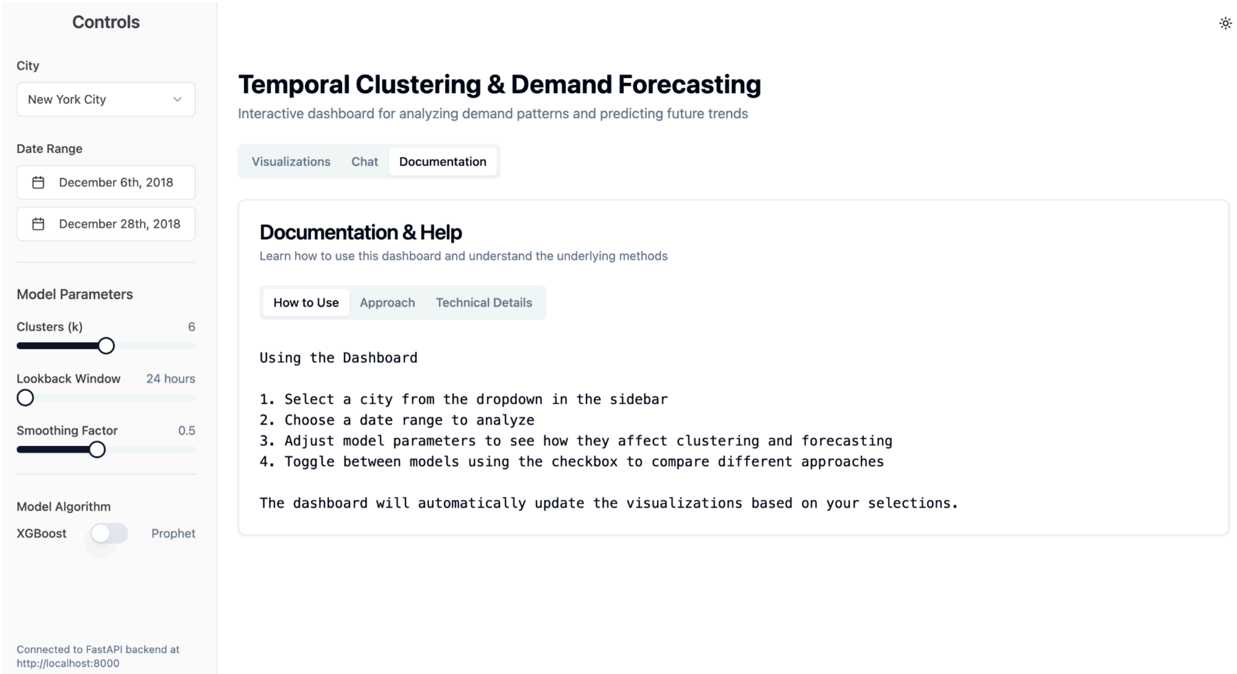
The electricity demand forecasting system comprises a comprehensive set of features designed to provide accurate predictions, insightful analysis, and user-friendly interaction.

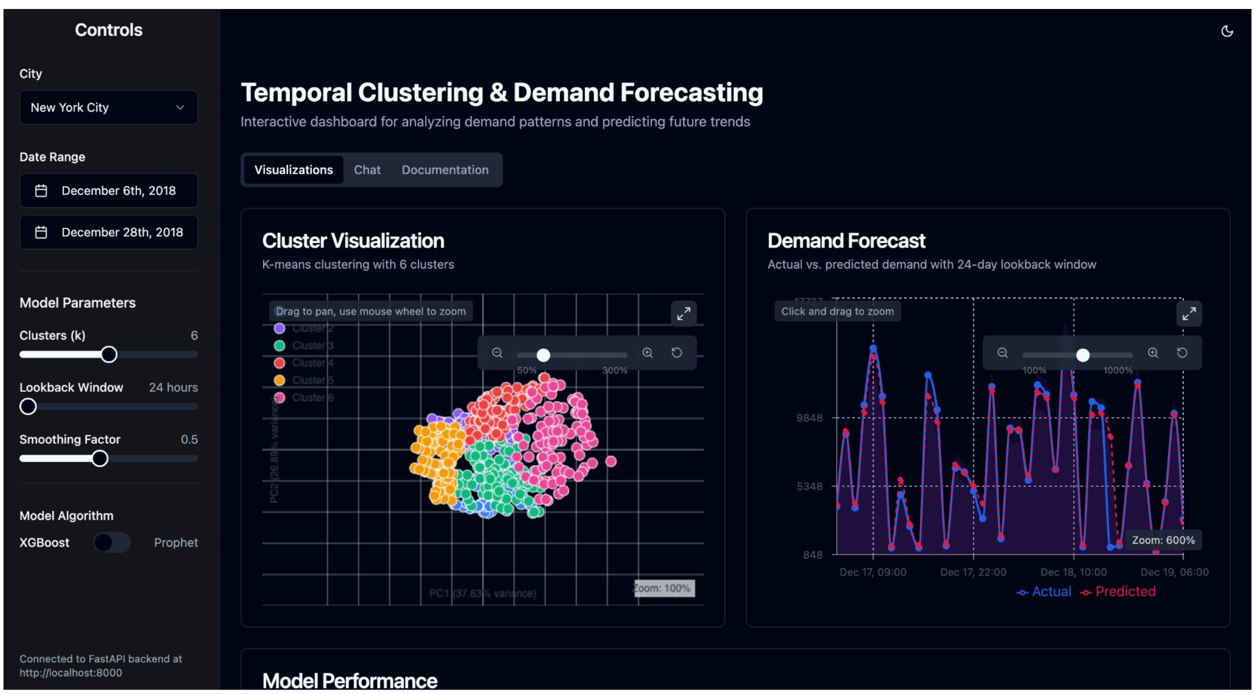
2.1 Features

- **Demand Forecasting:** The system employs machine learning models to predict future electricity demand based on historical data and relevant factors.
- **Clustering Analysis:** It identifies patterns in electricity usage through cluster analysis, enabling a deeper understanding of consumption behavior.
- **AI Assistant:** An integrated AI assistant provides users with explanations and insights about electricity forecasting, enhancing the interpretability of the results.
- **Interactive UI:** A modern web interface allows users to visualize data, explore forecasts, and interact with the system.
- **Real-time Analytics:** The system supports the processing and visualization of real-time data, providing up-to-date insights.




Controls
City
New York City
Date Range
December 6th, 2018
December 28th, 2018
Model Parameters
Clusters (k) 6
Lookback Window 24 hours
Smoothing Factor 0.5
Model Algorithm
XGBoost Prophet
Connected to FastAPI backend at http://localhost:8000
Temporal Clustering & Demand Forecasting
Interactive dashboard for analyzing demand patterns and predicting future trends
Visualizations
Chat
Documentation
AI Assistant
Ask questions about demand forecasting and clustering models
Tell me about the data used in this project
Okay, I can tell you about the data used in our electricity demand forecasting project.
The dataset covers a time range from **2018-07-01 to 2020-05-19**.
It includes electricity demand data for **10 specific cities**:
* la
* phoenix
* san_antonio
* seattle
* san_diego
* dallas
* san_jose
* houston
* nyc
Ask a question about demand forecasting...


Controls
City
New York City
Date Range
December 6th, 2018
December 28th, 2018
Model Parameters
Clusters (k)
6
Lookback Window
24 hours
Smoothing Factor
0.5
Model Algorithm
XGBoost
Prophet
Connected to FastAPI backend at http://localhost:8000
Temporal Clustering & Demand Forecasting
Interactive dashboard for analyzing demand patterns and predicting future trends
Visualizations
Chat
Documentation
Documentation & Help
Learn how to use this dashboard and understand the underlying methods
How to Use
Approach
Technical Details
Using the Dashboard
1. Select a city from the dropdown in the sidebar
2. Choose a date range to analyze
3. Adjust model parameters to see how they affect clustering and forecasting
4. Toggle between models using the checkbox to compare different approaches
The dashboard will automatically update the visualizations based on your selections.


Controls
City
New York City
Date Range
December 6th, 2018
December 28th, 2018
Model Parameters
Clusters (k)
6
Lookback Window
24 hours
Smoothing Factor
0.5
Model Algorithm
XGBoost
Prophet
Connected to FastAPI backend at http://localhost:8000
Temporal Clustering & Demand Forecasting
Interactive dashboard for analyzing demand patterns and predicting future trends
Visualizations
Chat
Documentation
Cluster Visualization
K-means clustering with 6 clusters
Drag to pan, use mouse wheel to zoom
Zoom: 100%
Demand Forecast
Actual vs. predicted demand with 24-day lookback window
Click and drag to zoom
9848
5348
848
Zoom: 600%
Dec 17, 09:00
Dec 17, 22:00
Dec 18, 10:00
Dec 19, 06:00
Actual
Predicted
Model Performance

2.2 Key Technologies

The system is built using a combination of cutting-edge technologies:

- **Backend:** FastAPI (Python), scikit-learn, TensorFlow, XGBoost
- **Frontend:** React, Vite, Tailwind CSS, Shadcn UI
- **Data Processing:** Pandas, NumPy, SciPy
- **Visualization:** Recharts, D3.js
- **AI Integration:** Google Gemini API

3. Technical Approach

The system's technical approach involves a series of steps, from data processing to model development and evaluation.

3.1 Data Processing Pipeline

The data processing pipeline is crucial for preparing the data for model training and analysis.

3.1.1 Data Collection

The system gathers electricity demand data from sources like the EIA930 dataset, combined with weather information from various cities.

3.1.2 Data Loading and Inspection

The data loading and inspection process includes:

- Standardizing city names across data sources.
- Converting timestamps to datetime objects.
- Merging data on city and timestamp.
- Validating data schema and integrity.

3.1.3 Missing Value Handling

Missing values are handled using a combination of techniques:

- Time-series features: Linear interpolation for short gaps, pattern-based imputation for longer gaps.
- Weather features: City-specific median imputation, forward fill for gradual changes.

3.1.4 Feature Engineering

Feature engineering involves creating new features to improve model performance:

- Temporal features: Cyclical encoding of time.
- Weather transformations: Scaling, non-linear transformations, and binned features.
- Lag features: Previous day/week demand and rolling statistics.

3.1.5 Anomaly Detection

Anomaly detection identifies unusual patterns in the data:

- Statistical methods: Z-score, IQR, and rolling z-score.
- Machine learning methods: Isolation Forest, One-class SVM, and LSTM-Autoencoder.
- Domain-specific rules.

3.1.6 Dimensionality Reduction

Dimensionality reduction techniques, such as PCA, are used for visualization and clustering analysis.

3.2 Model Training

The system employs several machine learning models to capture different aspects of electricity demand patterns.

3.2.1 Models Implemented

- XGBoost: Captures non-linear relationships and feature interactions.
- LSTM Neural Networks: Learns long-term dependencies in time series data.
- Ensemble Model: Combines predictions from multiple models.

3.3 Clustering Analysis

The clustering module identifies patterns in electricity consumption using unsupervised learning techniques.

3.3.1 Clustering Algorithms

- K-means Clustering: Groups similar demand patterns.
- DBSCAN: Identifies unusual demand patterns.
- Hierarchical Clustering: Provides nested grouping structures.

3.3.2 Dimensionality Reduction Techniques

Techniques like PCA, t-SNE, and UMAP are used to visualize and analyze high-dimensional data.

3.3.3 Cluster Interpretation

Cluster interpretation involves analyzing the characteristics of each cluster, including feature importance, temporal distribution, and actionable insights.

3.4 Model Development

The model development process includes problem formulation, data splitting, training, and validation.

3.4.1 Problem Formulation

The electricity demand forecasting problem is formulated as a supervised learning regression task.

- Forecasting Horizon: 24-hour ahead hourly forecasts, 7-day ahead daily forecasts, and next-hour forecasts.
- Target Variable: Hourly electricity demand in kWh.
- Feature Set: Calendar, weather, historical demand, and engineered features.

3.4.2 Data Splitting Strategy

- Chronological split: Training (70%), Validation (15%), Test (15%).
- Rolling window evaluation.

3.4.3 Training and Validation Strategy

- Temporal cross-validation.
- Hyperparameter tuning.
- Feature importance analysis.

3.4.4 Baseline Comparison

The system compares advanced models with naive forecasting baselines, including previous day, previous week, and historical average baselines.

4. Implementation

The system is implemented using a modular architecture, with separate components for data processing, model training, API development, and the user interface.

4.1 Installation

The installation process involves cloning the repository, creating a virtual environment, installing dependencies, and setting up the environment.

4.2 Running the Application

The application consists of a backend API and a frontend application. The API server is started using Python, and the frontend is launched using npm.

4.3 API Endpoints

The API provides endpoints for:

- Root endpoint: Provides basic API information.
- Prediction endpoint: Predicts electricity demand.
- Clustering endpoint: Performs clustering analysis.
- Chat assistant endpoint: Provides AI-powered assistance.

4.4 API Documentation

The system provides interactive API documentation using Swagger UI and ReDoc.

5. Models

The system utilizes several machine learning models for forecasting.

5.1 XGBoost

XGBoost is a gradient boosting framework that offers high performance, handles non-linear relationships, and provides feature importance.

5.2 LSTM Neural Network

LSTM networks are specialized for sequential data, capable of learning long-term dependencies and capturing seasonal patterns.

5.3 Ensemble Model

The ensemble model combines the strengths of multiple models to improve accuracy and robustness.

5.4 Evaluation Metrics

The models are evaluated using metrics such as RMSE, MAE, MAPE, and R^2.

6. Dataset

The project uses electricity demand data with various features, including timestamps, demand values, weather data, temporal features, and location information. The data is sourced from EIA930 datasets and weather data providers.

7. Development

The system is designed to be extensible and maintainable. New features can be added by defining schema models, creating service implementations, and adding route handlers. New models can be trained using the provided scripts.

8. Conclusion

This project delivers a robust and accurate electricity demand forecasting system with advanced features for data analysis and user interaction. The system's modular design, comprehensive data processing pipeline, and use of state-of-the-art machine learning models make it a valuable tool for energy providers and researchers.

References

While the provided text doesn't have formal citations, here are some relevant research areas and example papers that are commonly referenced in electricity demand forecasting:

- **XGBoost for Time Series Forecasting:**
  - Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).
- **LSTM Networks for Electricity Demand:**
  - Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.
  - Shi, X., Gao, H., Xu, L., & Wong, D. W. K. (2017). Short-term electricity load forecasting using stacked autoencoder and LSTM. *Energy*, *141*, 229-240.
- **Clustering for Load Profile Analysis:**
  - Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100-108.
- **Feature Engineering for Load Forecasting:**
  - Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- **Review of Electricity Demand Forecasting:**
  - Bedi, P., & Toshniwal, D. (2019). A systematic review of electricity demand forecasting techniques. *International Journal of Electrical Power & Energy Systems*, *107*, 255-281.