Electricity Usage Prediction

BY AHMED AYAZ, KRISH PATEL & DYLAN PATEL

Brief Process

1. About Project
2. Data Preparation
3. Data Analysis
4. Modeling and Prediction

About Project

INTRODUCTION OF THE PROJECT

Introduction

- **Objective:** Analyze the influence of weather conditions (temperature, snow, precipitation) on electricity usage across U.S. states from 2001 to 2024.
- **Goals:** Investigate correlations between environmental factors and electricity demand, and build predictive models for forecasting usage and pricing trends.
- **Outcome:** Develop insights into how weather impacts electricity consumption and create tools for future forecasting.

Key Questions

1. How do weather factors like temperature, snow, and precipitation influence electricity usage across different states?

2. What is the relationship between weather conditions and electricity pricing trends over time?

3. How accurately can future electricity usage and pricing be predicted based on historical weather data?

Data Preparation

DATA SOURCES AND PREPARATION

Electricity Usage & Price Dataset

- **Source:** Data was sourced from Kaggle Electricity Prices Dataset.
- **Size:** Covers electricity usage, pricing from 2001–2023 across all U.S. states.
- **Contents:** Includes state-wise price, electricity sales, revenue.
- **Purpose:** Designed to analyze trends, forecast electricity usage, and pricing patterns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85870 entries, 0 to 85869
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   year               85870 non-null  int64
 1   month              85870 non-null  int64
 2   stateDescription   85870 non-null  object
 3   sectorName         85870 non-null  object
 4   customers          59830 non-null  float64
 5   price              85870 non-null  float64
 6   revenue            85870 non-null  float64
 7   sales              85870 non-null  float64
dtypes: float64(4), int64(2), object(2)
memory usage: 5.2+ MB
```

City Information Dataset

- Sourced from U.S. Cities Data by SimpleMaps
- Provides geographic data (latitude, longitude) and population for each city
- Location data used to retrieve precise weather data per city
- Population data used to analyze electricity usage patterns in relation to city size

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31120 entries, 0 to 31119
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   city         31120 non-null  object
 1   city_ascii   31120 non-null  object
 2   state_id     31120 non-null  object
 3   state_name   31120 non-null  object
 4   county_fips  31120 non-null  int64
 5   county_name  31120 non-null  object
 6   lat          31120 non-null  float64
 7   lng          31120 non-null  float64
 8   population   31120 non-null  int64
 9   density      31120 non-null  float64
 10  zips         31118 non-null  object
 11  id           31120 non-null  int64
dtypes: float64(3), int64(3), object(6)
memory usage: 2.8+ MB
None
```

Weather Data Gathering

- **Data Extraction:** Weather data for all U.S. cities was fetched using the Meteostat Python library.
- **City-Level Aggregation:** Daily weather data was aggregated into monthly averages for each city.
- **State-Level Aggregation:** The monthly averages were further aggregated to the state level.
- **Data Integration:** The prepared weather data was merged with electricity usage datasets.

Data Cleaning

- Goal: Ensure accuracy and consistency in the datasets for reliable analysis.
  - 1. Electricity Usage Data:
    - - Handled missing values and standardized column names/formats.
    - - Filtered relevant columns for analysis (state, sector, price, etc.).
  - 2. Weather Data:
    - - Converted temperature to Fahrenheit, removed outliers, and ensured date consistency.
  - 3. Data Type Standardization:
    - - Ensured proper data types for all columns, particularly dates and numeric data.

Data Integration

- Datasets Integrated:
  - **Electricity Usage Data:** Monthly records of usage, pricing, and customer counts by state and sector.
  - **Weather Data:** Monthly averages of temperature, precipitation, and snowfall.
  - **Population Data:** Total population summaries for each state.
- Integration Process:
  - Merged datasets on common keys (state, year, and month) for cohesive analysis.
  - Enables exploration of relationships between electricity usage, weather, and population trends.

```
ass 'pandas.core.frame.DataFrame'>
geIndex: 13850 entries, 0 to 13849
a columns (total 12 columns):
  Column             Non-Null Count  Dtype
  ------             --------------  -----
  year               13850 non-null  int64
  month              13850 non-null  int64
  state              13850 non-null  object
  customers          9650 non-null   float64
  price              13850 non-null  float64
  revenue            13850 non-null  float64
  sales              13850 non-null  float64
  month_name         13850 non-null  object
  avg_temp           13850 non-null  float64
  precipitation      13850 non-null  float64
  snowfall           13748 non-null  float64
  total_population   13850 non-null  float64
pes: float64(8), int64(2), object(2)
ory usage: 1.3+ MB
```

Feature Engineering

- Purpose: Enhancing dataset with meaningful attributes to capture intricate patterns in electricity usage, weather, and pricing.
- Key Additions:
  - Time-based Features: Seasons and quarters for understanding seasonal trends.
  - Weather Features: Metrics like temperature range and categorized precipitation intensity.
  - Usage Metrics: Per capita electricity usage and price-to-sales ratio.
- Objective: Enable deeper insights into consumption trends and improve prediction model accuracy.

Final Dataset ready for analysis

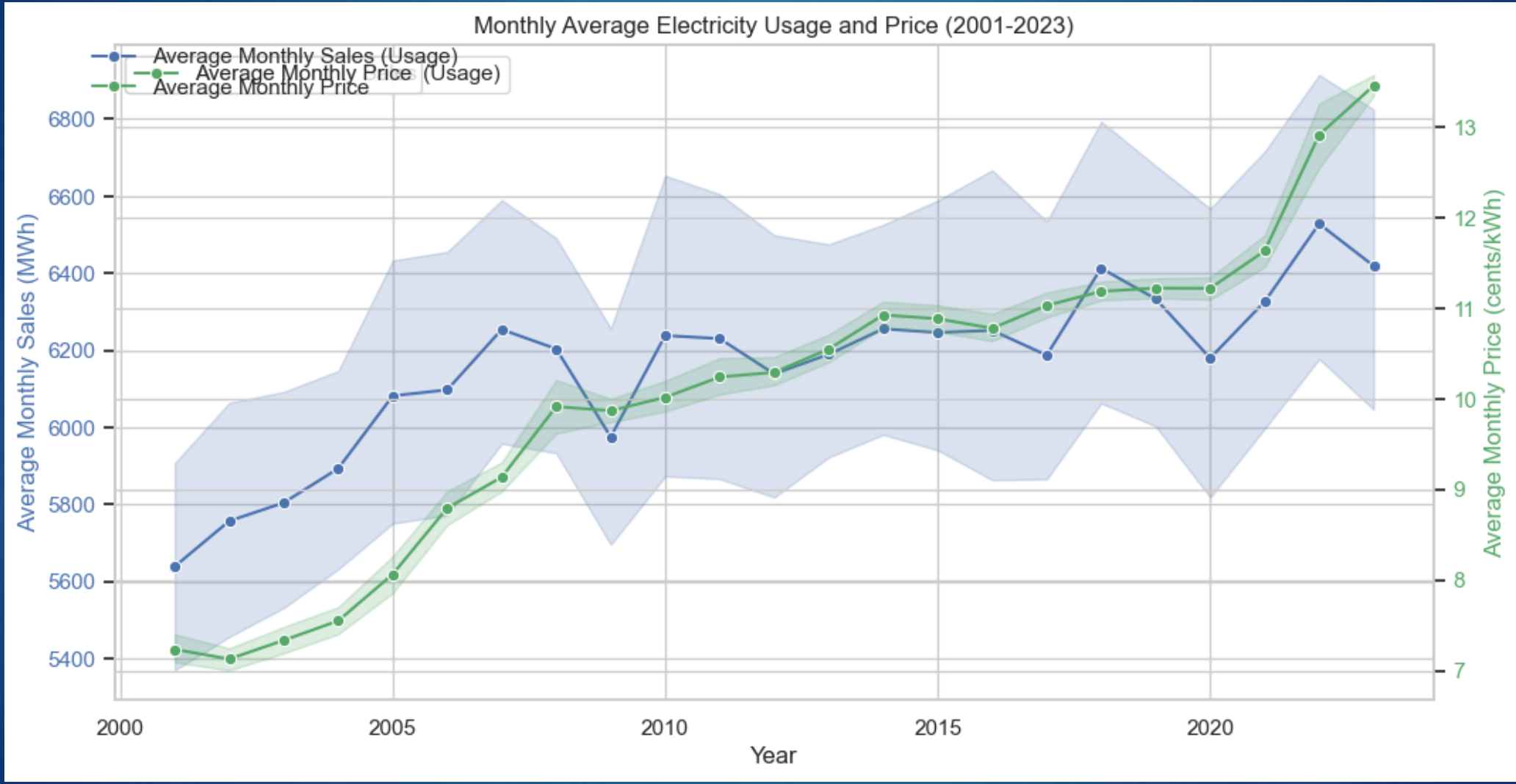
```
Data columns (total 26 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   year                    13850 non-null  int64
 1   month                   13850 non-null  int64
 2   state                   13850 non-null  object
 3   customers               9650 non-null   float64
 4   price                   13850 non-null  float64
 5   revenue                 13850 non-null  float64
 6   sales                   13850 non-null  float64
 7   month_name              13850 non-null  object
 8   avg_temp                13850 non-null  float64
 9   precipitation           13850 non-null  float64
 10  snowfall                13748 non-null  float64
 11  total_population        13850 non-null  float64
 12  season                  13850 non-null  object
 13  is_holiday_season       13850 non-null  bool
 14  is_summer_peak          13850 non-null  bool
 15  is_winter_peak          13850 non-null  bool
 16  is_high_temp            13850 non-null  bool
 17  is_low_temp             13850 non-null  bool
 18  has_precipitation       13850 non-null  bool
 19  is_high_precipitation   13850 non-null  bool
 20  has_snowfall            13850 non-null  bool
 21  is_heavy_snowfall       13850 non-null  bool
 22  usage_per_customer      9650 non-null   float64
 23  revenue_per_customer    9650 non-null   float64
 24  usage_yoy_growth        13250 non-null  float64
 25  price_yoy_growth        13250 non-null  float64
dtypes: bool(9), float64(12), int64(2), object(3)
```

Data Analysis

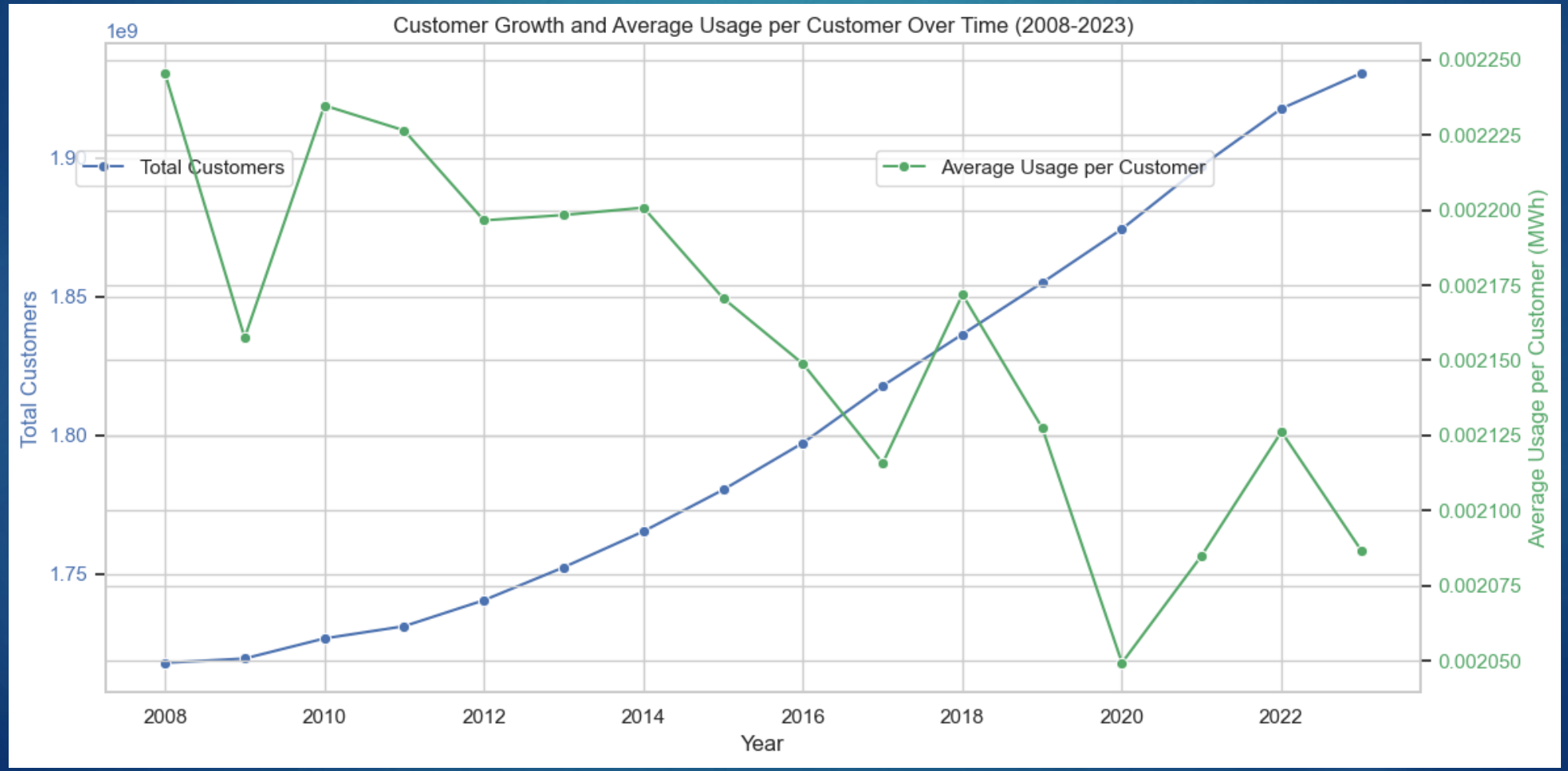
Exploratory Data Analysis

- Distribution of key features.
- Visualized trends and relationships in data.
- Identified missing data in key columns to assess data completeness.
- Temperature and population variability strongly influence electricity demand and pricing.

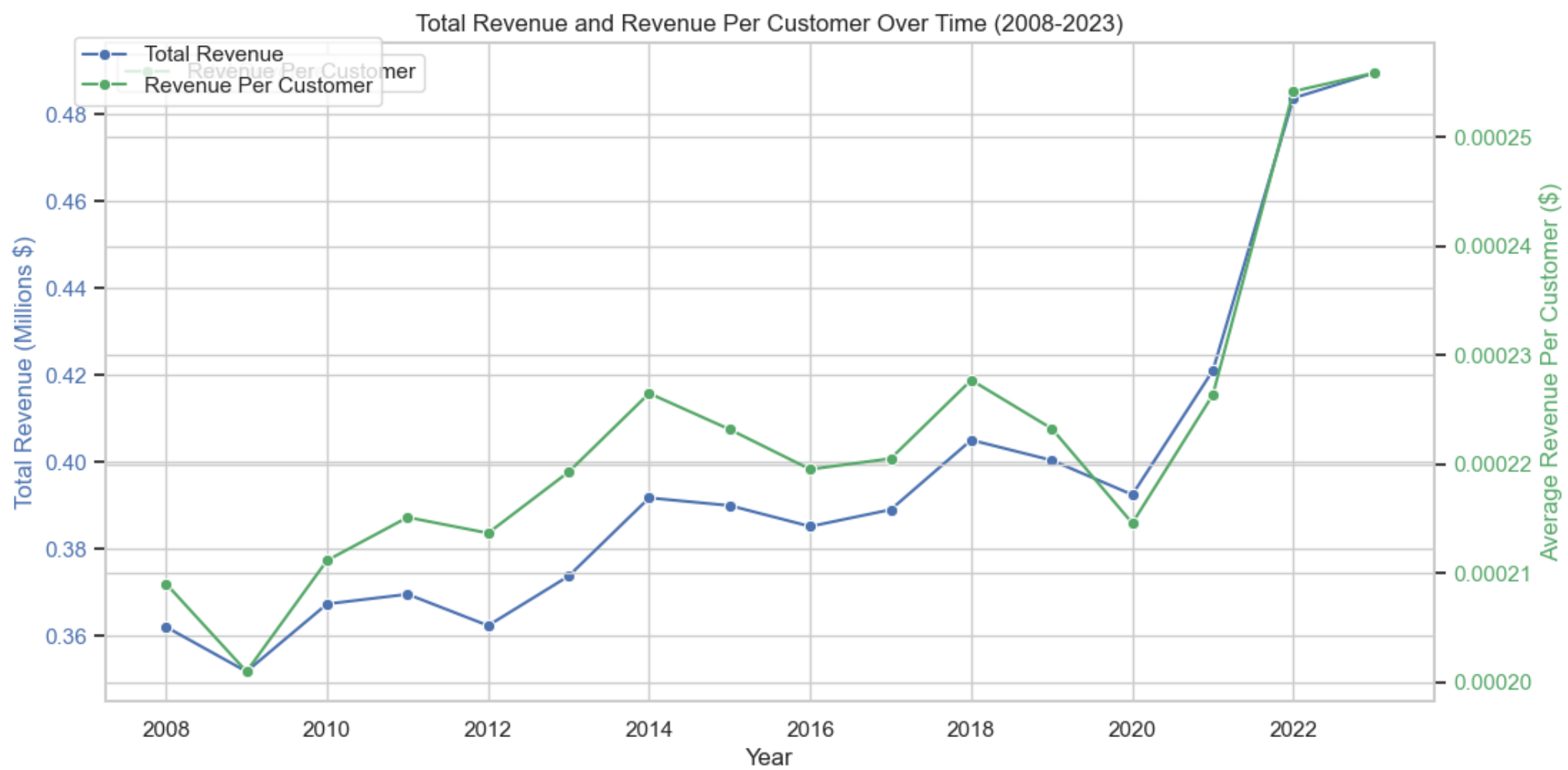
Electricity Usage & Price Over Time



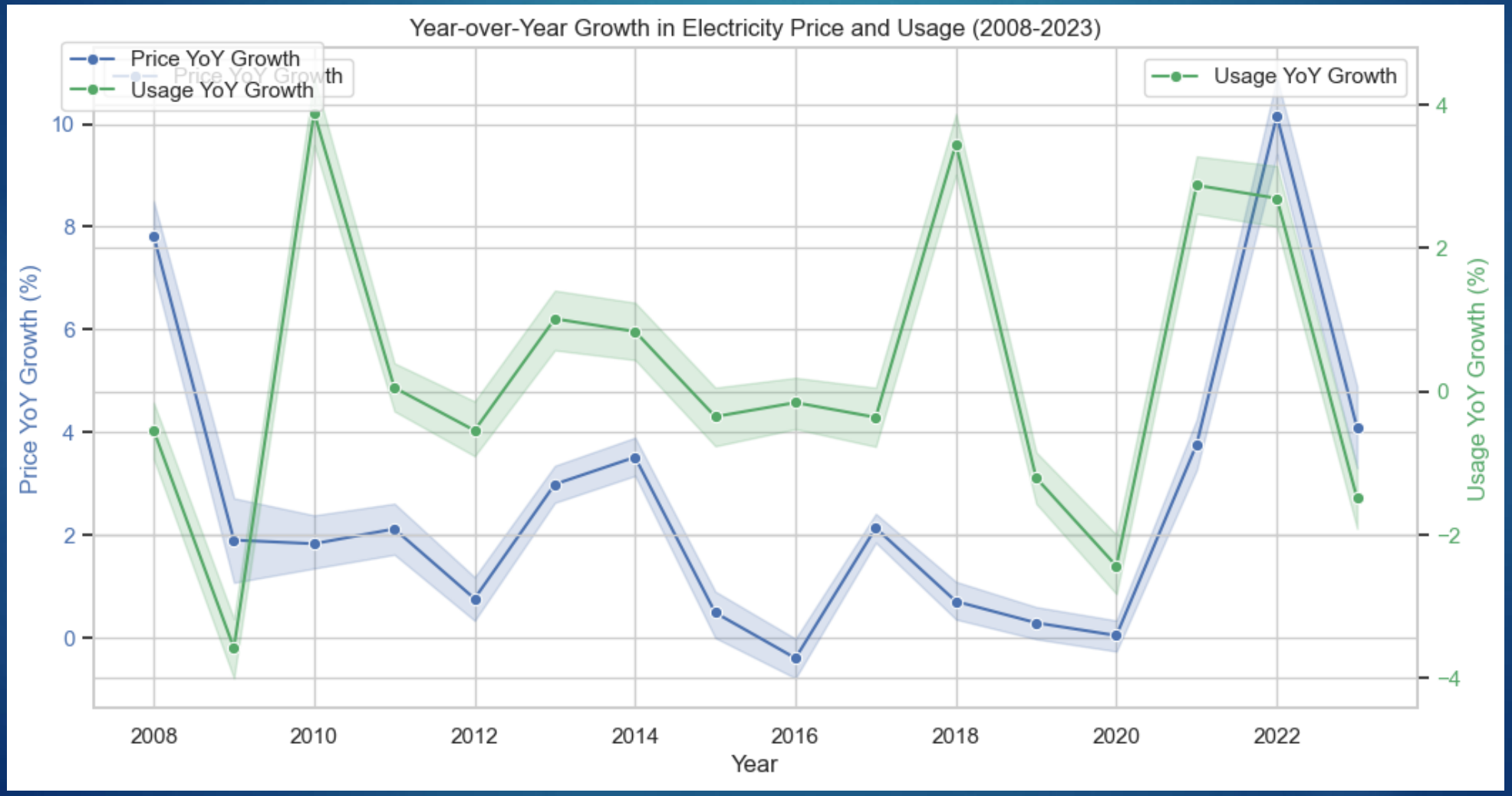
Customer Growth & Usage Per Customer



Total Revenue and Revenue Per Customer



Price YoY Growth and Usage YoY Growth



Modeling and Prediction

Modeling Selection

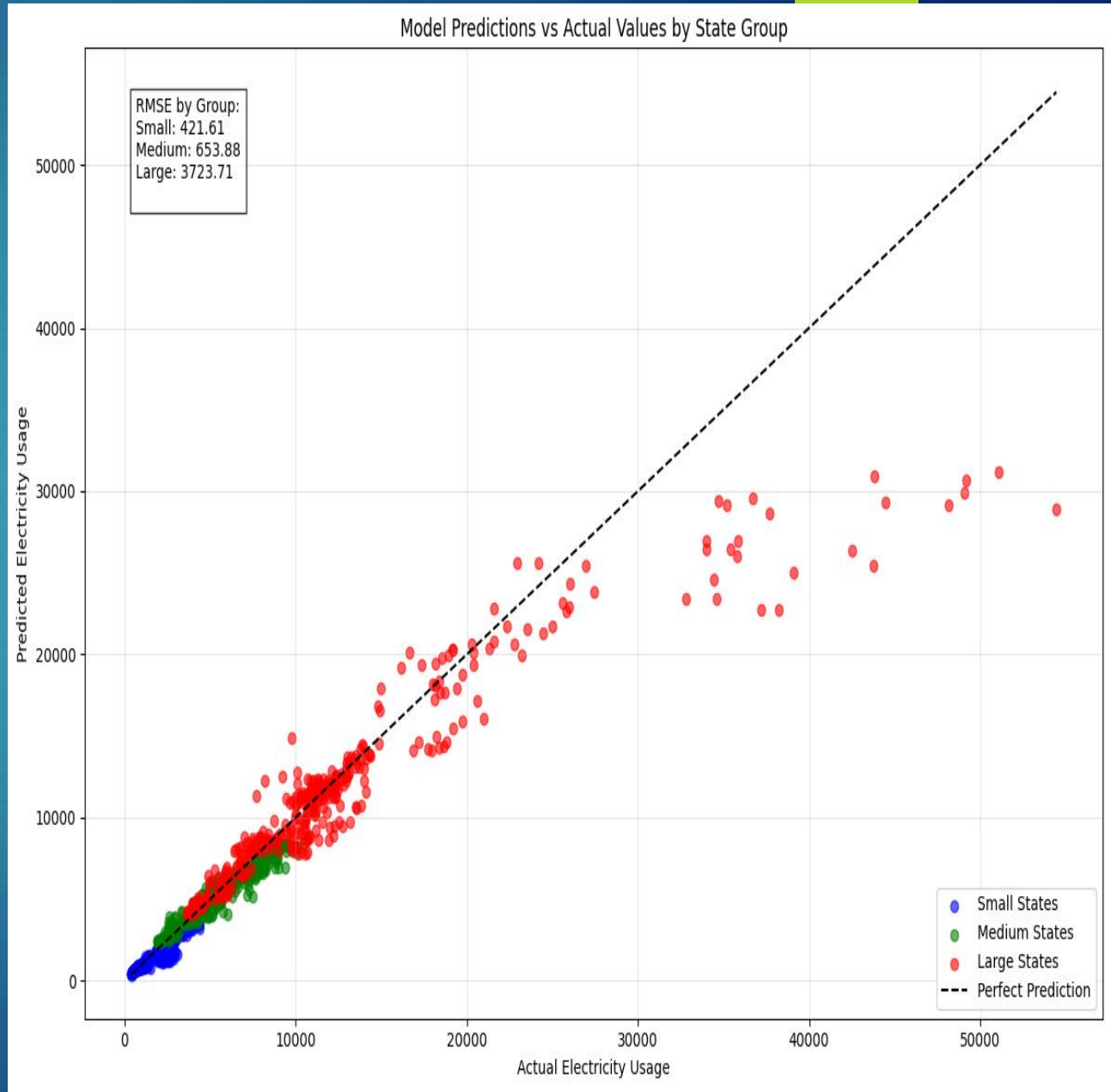
- XGBoost is based on Gradient Boosting and ensemble learning, combining multiple weak models to improve accuracy.
- Handles non-linear relationships between features and target, unlike linear regression.
- XGBoost can be thought of multiple experts working together
- Optimized for speed and efficiency with parallel processing.
- Provides insights into feature importance and allows for hyperparameter tuning.

Data Preparation and Feature Selection

- Key missing data identified: customers, usage_per_customer, revenue_per_customer.
- primary_features = [
- 'price', 'avg_temp', 'precipitation', 'total_population',
- 'season', 'is_summer_peak', 'is_winter_peak',
- 'is_high_temp', 'is_low_temp', 'has_precipitation',
- 'is_high_precipitation', 'has_snowfall', 'is_heavy_snowfall'
-]
- Features selected for the model (no missing data):
  - price, avg_temp, precipitation, total_population, season, is_summer_peak
- Target variable: 'sales' (electricity usage)
- Time-based data split (Train: 2008-2019, Validation: 2020-2021, Test: 2022-2023)

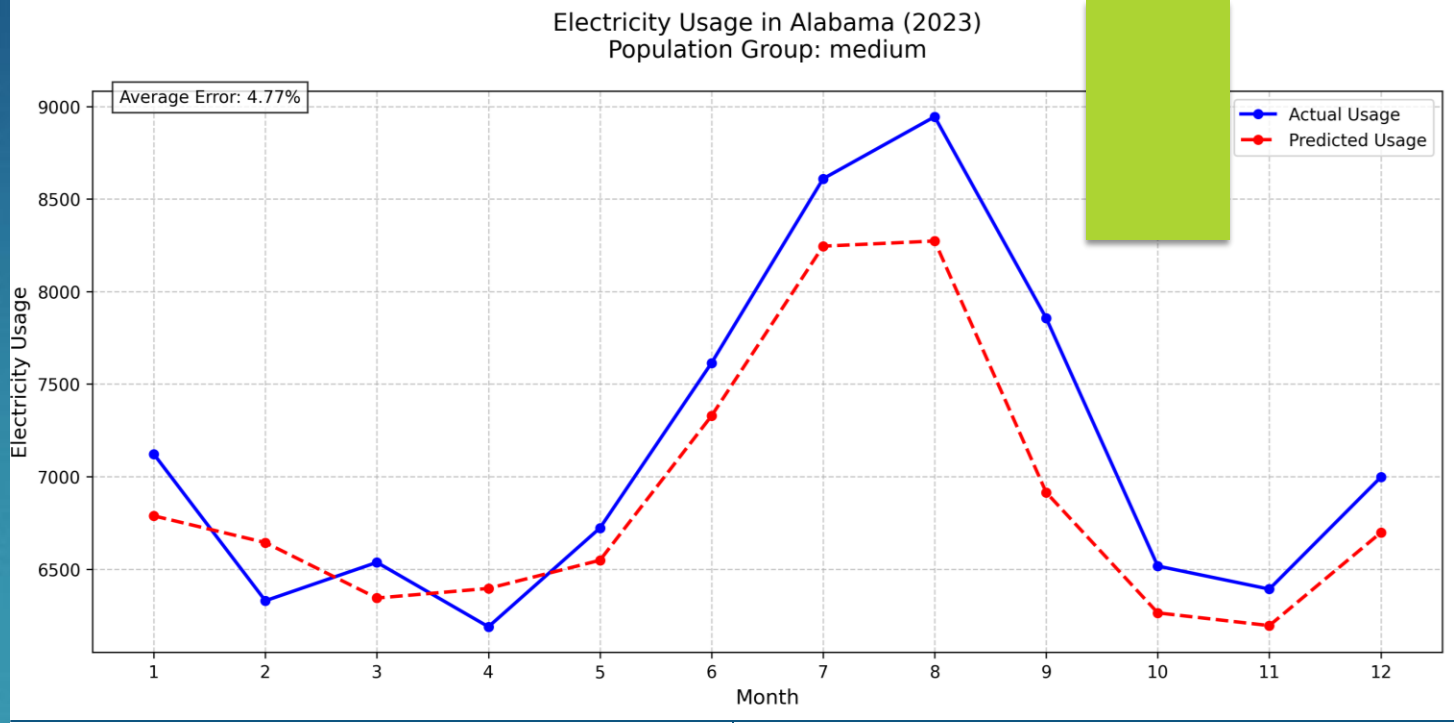
Model Creation and learning from mistakes

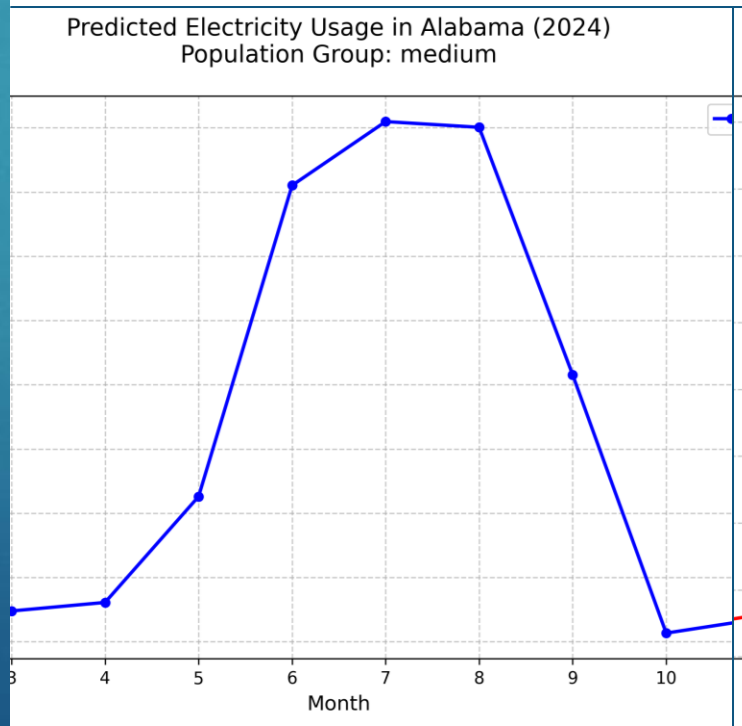
- Used a Regressor Model (XGBRegressor)
- First approach: Use **one model** for all states, Error Percentages:
  - Small size states: 9.8% ✅ Good
  - Medium size states: 3.6% ✅ Very Good
  - Large size states: 23.6% ❌ Poor
- Second approach: **"Log Transform"**
  - Small size states: 4.84% ✅ Very Good
  - Medium size states: 1.69% ✅ Excellent
  - Large size states: 27.37% ❌ Got even worse :(
- Third approach: "**Stratified Approach**" - Population-Based Strategy
  - Average Error for Large states dropped to **10%**

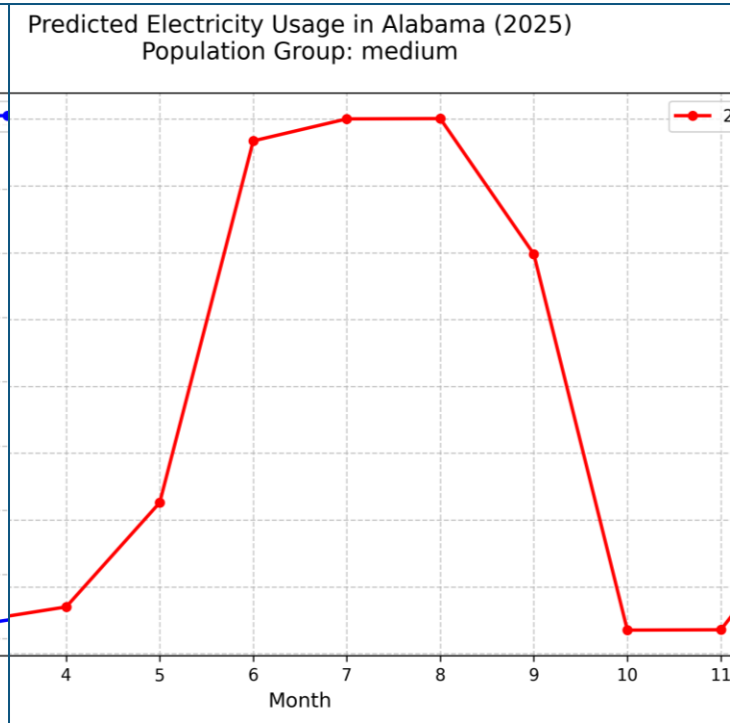


Predicting Future

- How did I predict future variables?
  - Average increase or decrease in temperature
  - Average increase or decrease in price per dollar
  - Locked at max ± 2 ($ or °F)
- After applying these changes, used our stratified model to predict years 2024 and 2025







References

Data Sources:

- Kaggle (Electricity Dataset) https://www.kaggle.com/datasets/alistairking/electricity-prices
- Meteostat (Weather Data) https://dev.meteostat.net/
- SimpleMaps (City Data) https://simplemaps.com/data/us-cities

Tools & Libraries:

- NumPy, Pandas, XGBoost
- Scikit-learn, Seaborn, Matplotlib
- Panel, Flask, Renderer


Q&A