**Week 1: Exploratory Data Analysis (EDA), Data Cleaning, and Initial Visualizations**

**What Was Done**

* **Objective: The goal was to explore and prepare the electricity load dataset for time-series forecasting, laying the groundwork for the 168-hour prediction project. The dataset, sourced from AEP\_hourly.csv, spanned January 10, 2004, to August 3, 2018, with 121,273 hourly records.**
* **Exploratory Data Analysis (EDA):** 
  + **Loaded the dataset using pandas and inspected its structure, confirming two columns: Datetime (object) and AEP\_MW (float64), with no missing values initially.**
  + **Analyzed summary statistics, noting a mean load of 15,499.51 MW, a standard deviation of 2,591.40 MW, and a maximum of 25,695 MW, which suggested potential outliers.**
  + **Identified the dataset’s suitability for time-series analysis due to its large size and temporal nature.**
* **Data Cleaning:** 
  + **Converted the Datetime column to a datetime format, finding no invalid entries.**
  + **Set Datetime as the index and sorted chronologically, resolving 4 duplicate timestamps by keeping the first occurrence.**
  + **Checked for missing AEP\_MW values (none found), avoiding the need for imputation at this stage.**
  + **Detected 259 outliers beyond 3 standard deviations (e.g., 23,558 MW on July 25, 2005, 14:00), opting to note them for later review rather than remove them initially.**
* **Initial Visualizations:** 
  + **Created a full time-series plot (2004-2018) with Plotly, revealing seasonal peaks in summer and winter, likely from air conditioning and heating demand.**
  + **Generated a one-week plot (January 1-7, 2005), showing daily cycles with peaks at 6-7 PM and lows at 3-5 AM, reflecting human activity patterns.**
  + **Produced a one-month plot (January 2005), highlighting a mid-month spike (January 15-20) possibly due to a cold snap.**
  + **Zoomed into an outlier period (July 25-26, 2005), confirming peaks around 14:00-16:00 (up to 24,000 MW) as heatwave-driven demand.**
  + **Saved all plots as HTML files (e.g., full\_time\_series.html)**
* **Data Saving:** 
  + **Saved the cleaned dataset as cleaned\_dataset.csv and cleaned\_dataset.pkl for future use, ensuring efficient reloading.**

**Technical Approach**

* **Used Python with pandas for data manipulation, plotly for interactive visualizations, and Jupyter Notebook for an iterative workflow.**
* **Managed file paths with os to organize outputs in PLOT\_DIR and PICKLE\_DIR.**
* **Handled large data with in-memory processing, noting the need for chunking if memory issues arose later.**

**Challenges Overcome**

* **Decided to retain outliers for initial analysis, planning to revisit their impact in Week 2 modeling.**

**Resources**

* **Code and Guidance: Relied on the EDA and visualization code you shared, refined with my input on April 3, 2025, including data loading, cleaning steps (e.g., pd.to\_datetime, outlier detection), and Plotly plotting.**
* **Learning Material: Leveraged pandas documentation for data handling and plotly tutorials for creating interactive time-series plots, likely supplemented by general EDA guides.**
* **Dataset: Started with the original AEP\_hourly.csv from your specified path.**

**Interview Talking Points**

* **Opening: "In Week 1, I began by exploring the electricity load dataset, a 14-year hourly record, to set up the project for a 168-hour forecast."**
* **EDA: "I analyzed the data’s structure and statistics, spotting a max load of 25,695 MW that hinted at outliers, confirming its time-series potential."**
* **Cleaning: "I converted timestamps, removed duplicates, and noted 259 outliers, deciding to keep them for now to preserve data integrity."**
* **Visualizations: "I created Plotly plots for the full period, a week, a month, and an outlier period, revealing seasonal and daily patterns tied to weather and usage."**
* **Saving: "I saved the cleaned data as CSV and pickle files, ensuring accessibility for future steps."**
* **Challenges: "I handled the missing kaleido issue by saving HTML plots and planned to address outliers later, keeping the process flexible."**
* **Resource Use: "I used pandas and plotly documentation, plus the code guidance you gave, to navigate EDA and visualization effectively."**

**Preparation Tips**

* **Practice describing a specific insight (e.g., "The July 2005 outlier peak at 24,000 MW suggested a heatwave, which I confirmed with the plot").**
* **Be ready to explain the outlier decision (e.g., "I kept them to avoid losing real events, planning to test their impact in modeling").**
* **Transition to Week 2 by noting how visualizations informed model selection.**

**Preparation Tips**

* Practice describing a specific insight (e.g., "The July 2005 outlier peak at 24,000 MW suggested a heatwave, which I confirmed with the plot").
* Be ready to explain the outlier decision (e.g., "I kept them to avoid losing real events, planning to test their impact in modeling").
* Transition to Week 2 by noting how visualizations informed model selection.

**Week 2: Feature Engineering, Exploratory Data Analysis (EDA), and Data Preparation**

**What Was Done**

* **Objective**: The focus was to enhance the dataset with meaningful features, conduct deeper exploratory analysis, and prepare it for modeling in the 168-hour electricity load forecasting project. This built on the cleaned data from Week 1, using the cleaned\_dataset.pkl file.
* **Feature Engineering**:
  + Added time-based features: Hour (0-23), DayOfWeek (0-6, where 0 is Monday), Month (1-12), Year, and IsWeekend (1 for Saturday/Sunday, 0 otherwise) to capture temporal patterns.
  + Noted a minor inconsistency in IsWeekend (e.g., 05:00 on a Saturday marked as 0) but proceeded with the feature set.
  + Introduced advanced features: a 1-hour lag (Lag\_1) to reflect prior load dependency, and seasonal indicators IsSummer (June-August) and IsWinter (December-February) based on observed peaks.
  + Saved the updated dataset as updated\_dataset.csv and updated\_dataset.pkl, then as advanced\_dataset.csv and advanced\_dataset.pkl after adding advanced features.
* **Exploratory Data Analysis (EDA)**:
  + Created a plot of average load by hour, revealing a peak of ~17,000 MW at 17:00-19:00, indicating high evening demand likely from residential use.
  + Generated a plot of average load by day, showing lower loads on weekends (~14,500-15,000 MW) compared to weekdays (~15,500-16,000 MW), suggesting reduced industrial activity.
  + Produced a plot of average load by month, highlighting higher loads in July-August (~17,000 MW) and December (~16,500 MW) due to summer cooling and winter heating, with a low in April (~14,000 MW).
  + Verified IsWeekend with 28.56% weekend hours, aligning with expected weekend patterns.
* **Data Preparation**:
  + Conducted a gap check, identifying 27 missing hours (e.g., October 31, 2004, 02:00) likely due to Daylight Saving Time (DST) transitions or data collection gaps.
  + Addressed a single missing Lag\_1 value (initial row) with no immediate imputation, noting it for modeling consideration.
  + Saved the final dataset as final\_dataset.csv and final\_dataset.pkl, ready for Week 3 modeling.

**Technical Approach**

* Utilized Python with pandas for feature engineering and data manipulation, and plotly for interactive visualizations.
* Managed file paths with os to organize outputs in PLOT\_DIR and PICKLE\_DIR.
* Used Jupyter Notebook for iterative analysis and visualization, ensuring data integrity with each save.

**Challenges Overcome**

* Resolved the IsWeekend inconsistency by accepting it as a minor edge case (e.g., early morning hours), planning to revisit if it impacted modeling.
* Handled the 27 missing hours by documenting them, preparing to address DST effects or imputation in future steps.
* Managed the missing Lag\_1 value by leaving it as-is, anticipating model robustness to handle it.

**Resources**

* **Code and Guidance**: Built on the feature engineering and EDA code you provided, refined with my input from our April 8-12, 2025, discussions, including feature extraction (e.g., df.index.hour) and Plotly plotting.
* **Learning Material**: Referenced pandas documentation for datetime features and plotly tutorials for line plots, likely supplemented by time-series EDA guides I’ve previously shared (e.g., from our stock market and climate data talks on February 22 and 24, 2025).
* **Dataset**: Started with cleaned\_dataset.pkl from Week 1, evolving it through updated\_dataset.pkl and advanced\_dataset.pkl to final\_dataset.pkl.

**Interview Talking Points**

* **Opening**: "In Week 2, I enriched the dataset with features and conducted deeper EDA to prepare for the 168-hour load forecast, building on Week 1’s foundation."
* **Feature Engineering**: "I added time features like hour and day, plus a lag and seasonal indicators for summer and winter, saving each iteration to track progress."
* **EDA**: "I plotted average loads by hour, day, and month, spotting evening peaks, weekend dips, and seasonal highs, which guided my modeling approach."
* **Data Preparation**: "I identified 27 missing hours, likely from DST, and handled a missing lag value, ensuring the dataset was ready for modeling."
* **Challenges**: "I noted the IsWeekend inconsistency and missing data, planning to address them later, while keeping the process adaptable."
* **Resource Use**: "I used pandas and plotly documentation, plus the code guidance you gave, to enhance the dataset and visualize patterns effectively. Our past talks on time-series features also helped."

**Preparation Tips**

* Practice explaining a feature’s impact (e.g., "The Lag\_1 feature, with a mean close to the current load, suggested strong autocorrelation for forecasting").
* Be ready to discuss the gap check (e.g., "The 27 missing hours align with DST changes, which I’d impute if needed for continuity").
* Transition to Week 3 by mentioning how these features informed model selection.

**Week 3: Gap Handling, Stationarity Analysis, Modeling, Forecasting, and Visualization**

**What Was Done**

* **Objective**: The goal was to address data gaps, assess stationarity, develop and compare time-series models, generate 168-hour forecasts, and create visualizations to complete the electricity load forecasting project, using the final\_dataset.pkl from Week 2.
* **Gap Analysis and Handling**:
  + Identified and reindexed the dataset to fill 27 missing hours (e.g., October 31, 2004, 02:00), initially detected in Week 2, using pd.date\_range with hourly frequency.
  + Interpolated AEP\_MW values linearly and recalculated derived features (Hour, DayOfWeek, etc.) to resolve 27 missing values in those columns, leaving one Lag\_1 NaN.
  + Saved the interpolated dataset as interpolated\_dataset.csv and interpolated\_dataset.pkl.
* **Stationarity and Baseline Modeling**:
  + Conducted an Augmented Dickey-Fuller (ADF) test, yielding a p-value of 2.34e-30 and an ADF statistic of -18.25, indicating stationarity (p < 0.05).
  + Performed a KPSS test, with a statistic of 6.93 and p-value of 0.01, suggesting non-stationarity (p > 0.05), highlighting a discrepancy with ADF due to trends/seasonality.
  + Applied a 7-day (168-hour) moving average (MA\_7) to smooth the data.
  + Fitted an initial ARIMA(1,1,1) model with differencing (d=1) based on KPSS, achieving an AIC of 1,747,740 and a residual standard deviation of 327.54.
* **Advanced Modeling**:
  + Implemented SARIMA(1,1,1)x(1,1,1,24) with low\_memory=True, reducing AIC to 1,602,426 and residual standard deviation to 183.81, capturing daily seasonality.
  + Applied Exponential Smoothing with additive trend and seasonality (period=24), yielding an AIC of 1,385,098 despite a convergence warning, and added fitted values (ES\_Smoothed).
  + Detected volatility clustering with an ACF of squared SARIMA residuals (lag 24 = 0.181), confirming the need for a volatility model.
  + Fitted a GARCH(1,1) model on scaled data (AEP\_MW / 1000), achieving an AIC of -70.37 and a residual standard deviation of 0.259, with volatility rescaled to 1,000-1,500 MW.
* **Forecasting**:
  + Generated 168-hour forecasts (August 3-10, 2018) for ARIMA, SARIMA, and Exponential Smoothing, and volatility forecasts with GARCH.
  + Stored results in df\_forecast with columns ARIMA\_Forecast, SARIMA\_Forecast, ES\_Forecast, and GARCH\_Variance, saved as forecast\_results.csv and forecast\_results.pkl.
* **Visualization**:
  + Created a corrected plot with matplotlib, displaying load forecasts (ARIMA, SARIMA, ES) and GARCH Volatility (1,000-1,500 MW), aligning with the 12,000-17,000 MW load scale.
  + Compiled a final summary in final\_summary.txt, highlighting ES as the best mean model (AIC: 1,385,098), SARIMA’s stability (residual std dev: 183.81), and GARCH’s volatility capture.

**Technical Approach**

* Used Python with pandas for data handling, statsmodels for ARIMA, SARIMA, and Exponential Smoothing, arch for GARCH, and matplotlib for visualization.
* Managed large datasets with low-memory options and saved iterative outputs (e.g., interpolated\_dataset.pkl, smoothed\_dataset.pkl) in PICKLE\_DIR.

**Challenges Overcome**

* Resolved the gap handling oversight by recalculating derived features post-interpolation, ensuring data continuity.
* Addressed the ADF/KPSS discrepancy by forcing ARIMA with d=1 and enhancing with SARIMA, aligning with observed seasonality.
* Fixed GARCH volatility scaling (previously 0-500 MW) by adjusting the rescaling factor to 1,000, improving model relevance.

**Resources**

* **Code and Guidance**: Built on the modeling and visualization code you provided, refined with my input from April 15-17, 2025, including gap handling, stationarity tests, and GARCH adjustments.
* **Learning Material**: Referenced statsmodels documentation for ADF, KPSS, SARIMA, and Exponential Smoothing, and arch documentation for GARCH, supplemented by time-series modeling tutorials.
* **Dataset**: Started with final\_dataset.pkl, evolving through interpolated\_dataset.pkl, smoothed\_dataset.pkl, to volatility\_dataset.pkl and enhanced\_dataset.pkl.

**Interview Talking Points**

* **Opening**: "In Week 3, I finalized the project by handling gaps, modeling the data, generating forecasts, and visualizing results for the 168-hour load prediction."
* **Gap Handling**: "I filled 27 missing hours with interpolation and recalculated features, ensuring a continuous 121,296-row dataset."
* **Modeling**: "I confirmed non-stationarity with KPSS, fitted ARIMA (AIC: 1,747,740), improved with SARIMA (AIC: 1,602,426), and used ES (AIC: 1,385,098) despite convergence issues."
* **Volatility and Forecasting**: "I detected volatility clustering, fitted GARCH(1,1) with rescaled volatility (1,000-1,500 MW), and forecasted 168 hours, saving the results."
* **Visualization**: "I created a corrected plot showing all forecasts and volatility, with ES and SARIMA reflecting daily cycles, and compiled a summary for insights."
* **Challenges**: "I resolved the gap issue, adjusted GARCH scaling, and managed the convergence warning, ensuring robust outputs."
* **Resource Use**: "I used statsmodels and arch documentation, plus your guidance, to navigate modeling and visualization effectively."

**Preparation Tips**

* Practice explaining the stationarity conflict (e.g., "The ADF suggested stationarity, but KPSS and trends led me to use d=1, validated by SARIMA’s improvement").
* Be ready to discuss GARCH tuning (e.g., "The low beta[1] suggests I could try GARCH(1,2) for better persistence").
* Link to the project’s end by noting the dashboard transition in our later work.