Weathering the Storm: Forecasting Energy and Pricing Trends

This project aims to analyze energy consumption and consumption and pricing trends based on weather weather patterns and climate factors.

Data Engineering Pipeline

Data Ingestion

Data Explore & Transformation

Data Analysis

Detailed Steps

1. Data download and preprocessing
2. Null Values Check
3. Find Correlations
4. Plot Correlation Heatmap
5. Analyse Actual price of Energy
6. Data Wrangling and Weather Feature engineering
7. Plot - Price Per KW Hour Compared To Total Energy Generated Per Season

Data Exploration findings

- Normal distrubution of target variable.
- Low percentage of nan values in columns without 100% nan.
- Lots of columns with 0 values for certain energy generation types.
- Found very high correlation between some columns.
- All data seems to be numeric.

Data Cleaning Findings

- Removed only row from 2014 for plotting purposes.
- Removed forecasted columns to prevent data leakage.
- Removed all columns with 0 fill for all values.

Prefect Overview

Data Engineering Framework

- Prefect is an open-source workflow management system for Python that enables data engineer ETL (Extract, Transform, Load), and automation of complex processes.
- Designed for simplicity and flexibility, Prefect offers a Python-native interface for defining, sched and orchestrating workflows.
- Comparison with traditional tools: Prefect provides a more Pythonic, code-centric approach com to other workflow management systems, allowing for greater transparency and control.

Prefect Integration and Components

- **Jupyter Integration:** Prefect seamlessly integrates with Jupyter notebooks, enabling the creat and execution of workflows directly within the familiar Jupyter environment.
- **Flows:** Flows in Prefect represent the entire workflow, consisting of interconnected tasks. The provide a visual representation of the workflow logic and dependencies.
- **Tasks:** Tasks are the building blocks of Prefect workflows. Each task represents a unit of work can be a Python function, a shell command, or any other executable.

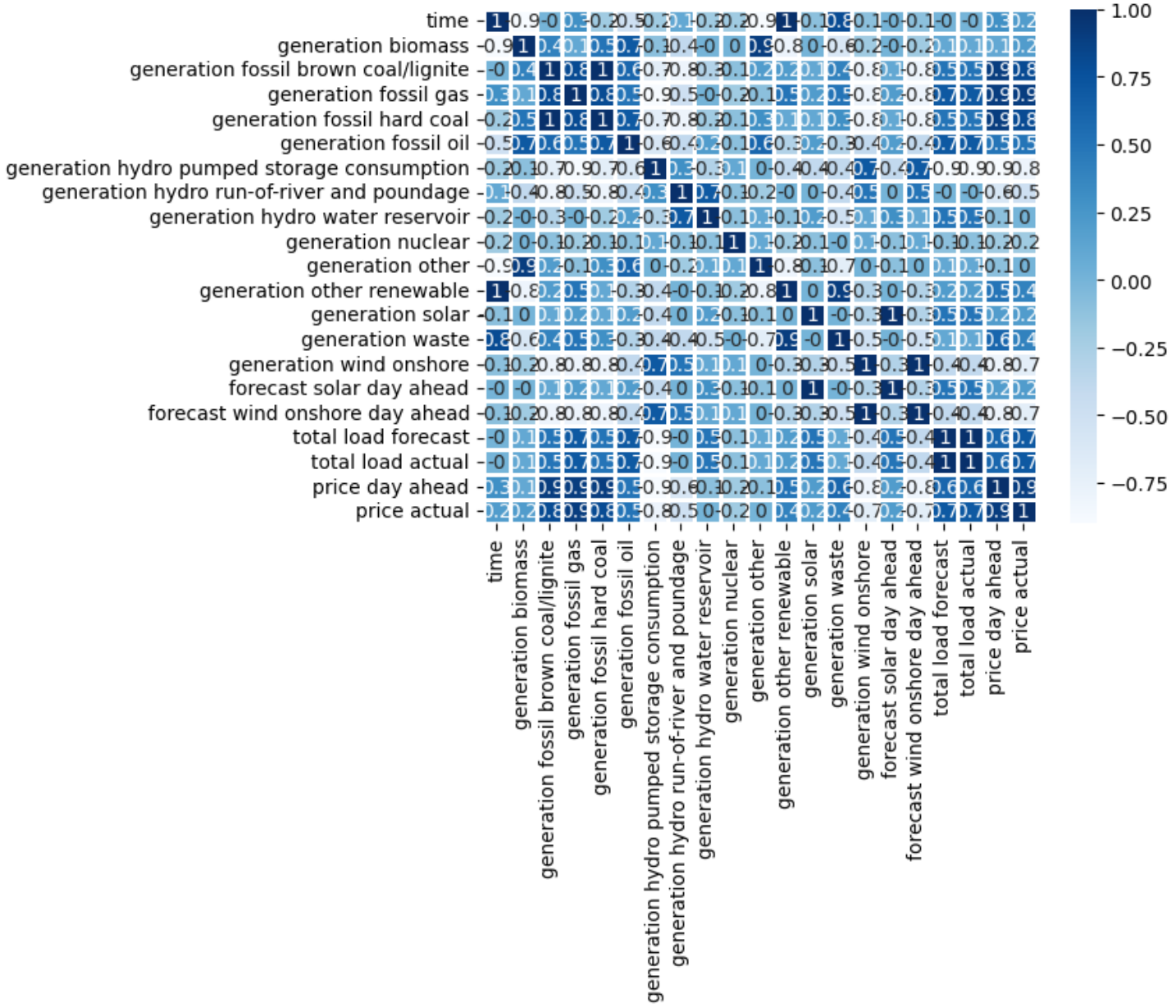
Logging and Visualization in Prefect

- **Logging:** Prefect offers comprehensive logging to track the execution of workflows. Logs prov insights into task execution, errors, and overall workflow performance.
- **Visualization:** Prefect provides a web-based UI for visualizing and monitoring workflows. The U allows users to explore flow runs, view task dependencies, and inspect individual task logs.
- **Comparison with Traditional Logging:** Compared to traditional logging in scripts or notebook Prefect's logging and visualization tools offer a more organized and user-friendly way to underst workflow execution.
- **Tasks:** Tasks are the building blocks of Prefect workflows. Each task represents a unit of work can be a Python function, a shell command, or any other executable.

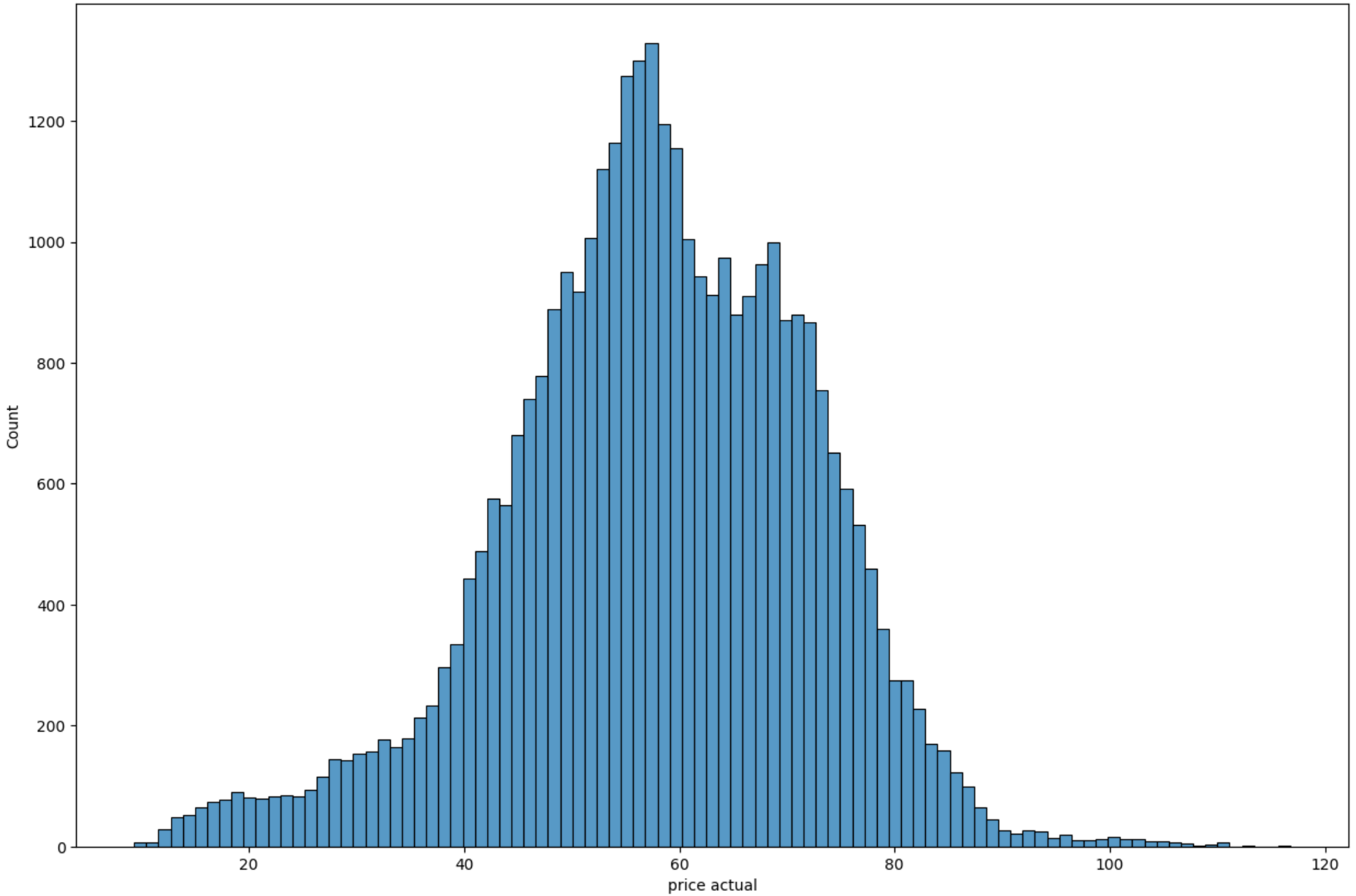
Deployment Strategies for Prefect Workflows

- **Local Deployment:** Workflows can be executed locally during development and testing using Prefect CLI or Python API.
- **Cloud Deployment:** Prefect supports deployment to cloud services such as AWS, Azure, and G Cloud. Cloud deployment allows for scalability and distributed execution.
- **Containerization:** Workflows can be containerized using tools like Docker, enabling consisten execution across different environments.
- **Scheduler Integration:** Prefect integrates with popular schedulers like Celery, Dask, and Kube for efficient and scalable task execution.

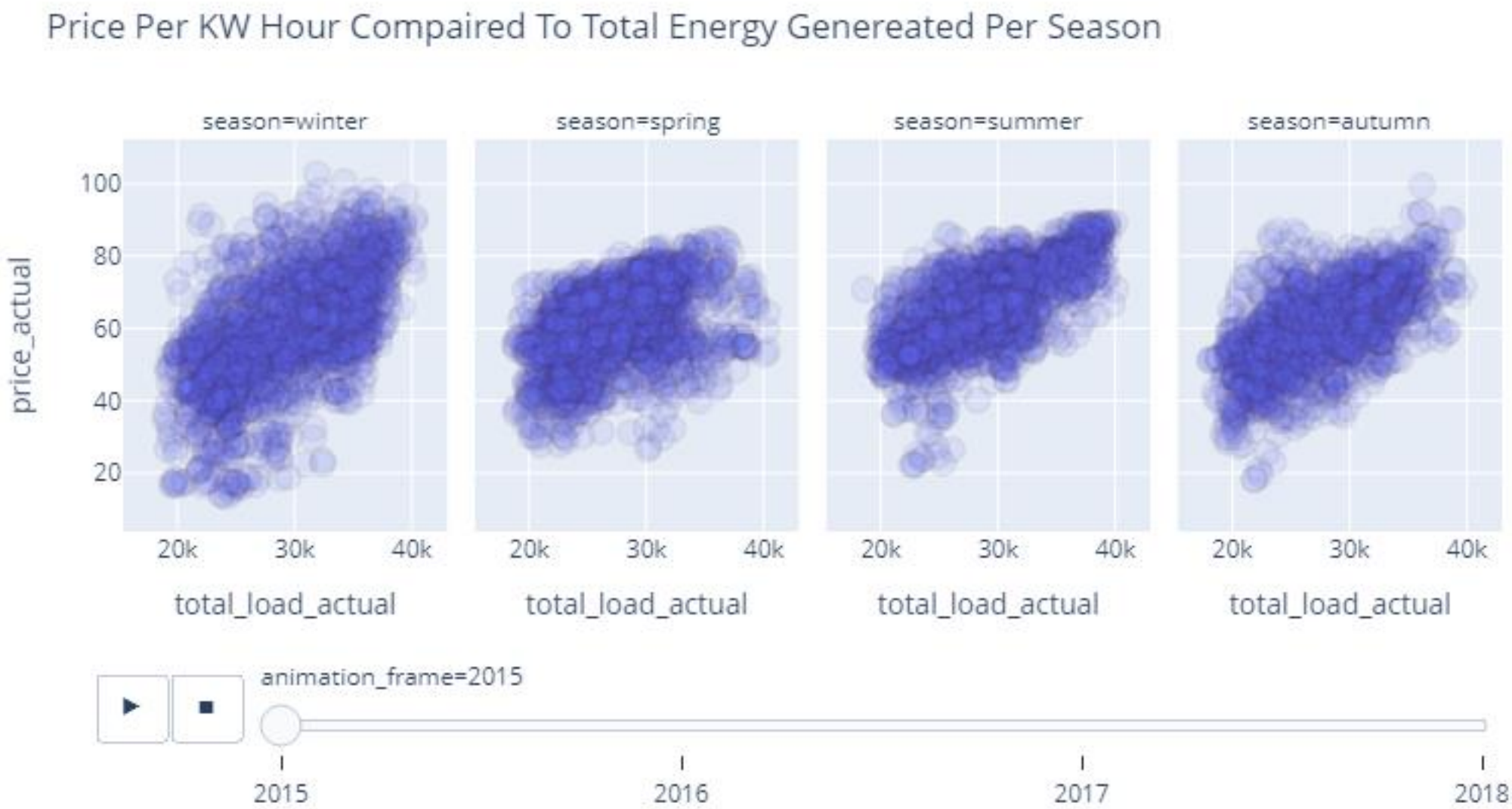
Correlation Heatmap



Price Actual Histogram



Price Per KW Season Wise



Conclusion

By harnessing the power of machine learning, this project will equip stakeholders with the knowledge and tools to navigate the evolving energy landscape and ensure energy security for all.

Proactively prepare for demand fluctuations caused by extreme weather events.

Optimize energy pricing strategies to ensure financial sustainability while maintaining affordability for consumers.