Household energy consumption

Author: ansem.cb@gmail.com
Start Date: Jul 22, 2024

1. Introduction

This project focuses on predicting the "active_power" column in a dataset that combines household energy consumption data with weather data from a single residence in Mexico. The data granularity is one-minute intervals. The project will include the following tasks:

- Data exploration and feature engineering will be used to prepare the dataset.
- Model development and training to predict "active_power".
- Containerization of the solution for easy deployment.
- The model is served using a web framework to provide predictions based on the input data.
- Proposing a cloud infrastructure schema for deployment.
- Discussing considerations for scalability, orchestration, maintenance, and model drift.

2. Data Overview

The dataset used in this project combines household energy consumption data with weather data from a single residence in Mexico over 14 months. The data is recorded at one-minute intervals, providing a detailed view of energy usage and corresponding weather conditions. The dataset source is publicly available on Mendeley Data: Household Energy Consumption.

The dataset encompasses a variety of data points related to household energy consumption and weather conditions:

- **Timestamp**: Records the date and time of each data point.
- **Energy Metrics**:
  - **Active Power**: Active power consumed by the residence.
  - **Current**: Electrical current flowing through the system.
  - **Voltage**: Electrical potential difference in the system.
  - **Reactive Power**: Reactive power in the system.
  - **Apparent Power**: Total power in the system, combining both active and reactive power.
  - **Power Factor**: Ratio of active power to apparent power.
- **Weather Conditions**:
  - **Main**: Main categorical weather condition.
  - **Description**: Detailed categorical weather conditions.
  - **Temp**: Ambient temperature at the time of recording.

    - **Feels Like**: Perceived temperature considering wind and humidity.
    - **Temp Min**: Minimum temperature recorded during the interval.
    - **Temp Max**: Maximum temperature recorded during the interval.
    - **Pressure**: Atmospheric pressure.
    - **Humidity**: Relative humidity.
    - **Speed**: Wind speed.
    - **Deg**: Wind direction in degrees.
- **Forecasted Weather**:
    - **Temp_t+1**: The forecasted temperature for the next day.
    - **Feels Like_t+1**: Forecasted 'feels like' temperature for the next day.

3. Exploratory Data Analysis (EDA)

For the Exploratory Data Analysis (EDA), I utilized a Streamlit app to effectively share and deploy my findings. This interactive app allowed me to visualize and explore the dataset dynamically, facilitating a deeper understanding of the data's characteristics and patterns.

Insights:

- **Data Distribution:**

  The distribution of active power exhibits a right-skewed pattern, characterized by a long tail on the right side. This indicates that most of the energy consumption values are concentrated at the lower end of the scale, with a smaller number of instances having significantly higher values.

- **Weather Impact**:

The weather in Mexico is predominantly clear with occasional clouds, and instances of rain and thunderstorms are relatively rare:

Weather Patterns: The predominant clear weather contributes to a lower frequency of extreme weather conditions that typically drive higher energy consumption. Thus, most of the data points are associated with lower active power values.

Impact of Rain and Thunderstorms: Rain and thunderstorms, although infrequent, are associated with significantly higher energy consumption. These weather conditions can lead to increased use of energy for cooling, lighting, or other reasons, which explains the higher active power values observed during these events.

- **Feature Relationships**:

  analysis reveals a high correlation (>0.9) between active_power, current, and apparent_power. A high correlation between active_power and current is expected

because, as the current increases, the active power (which is a function of current and voltage) also increases, assuming a constant voltage.

4. Feature Engineering

The columns `temp_t+1` and `feels_like_t+1` were excluded from the dataset.

The columns `current` and `apparent_power` were excluded from the dataset, due to their high correlation with the target.

Seasonality Features: Created lag features (e.g., `active_power_lag_1`, `active_power_lag_8`) to capture temporal dependencies and trends in the `active_power` series. These features help the model learn from past values to predict future ones.

DateTime Features: Added features such as `year`, `month_of_year`, `week_of_year`, `day_of_year`, `day_of_week`, `hour_of_day`, and `is_weekend` to provide the model with information about time-based patterns and seasonality.

Holiday Features: Introduced a binary feature `is_holiday` to indicate whether a given date is a holiday. This accounts for potential changes in energy consumption patterns during holidays.

Rolling and Expanding Window Features: Generated features based on rolling and expanding statistics (mean and standard deviation) over different time windows (e.g., daily, weekly, yearly). These features capture trends and variability in the `active_power` over different periods.

Scaling: Applied `StandardScaler` to numerical features to standardize their values, ensuring that they have a mean of 0 and a standard deviation of 1. This step improves the model's performance by making the features more comparable and reducing potential biases due to differing scales.

Encoding: Used `OneHotEncoder` to convert categorical features into numerical format, creating binary columns for each category. This transformation is necessary for incorporating categorical variables into machine learning models, which typically require numerical input.

5. Model Development

Various models (ARIMA, Prophet) were considered for predicting the active_power feature.

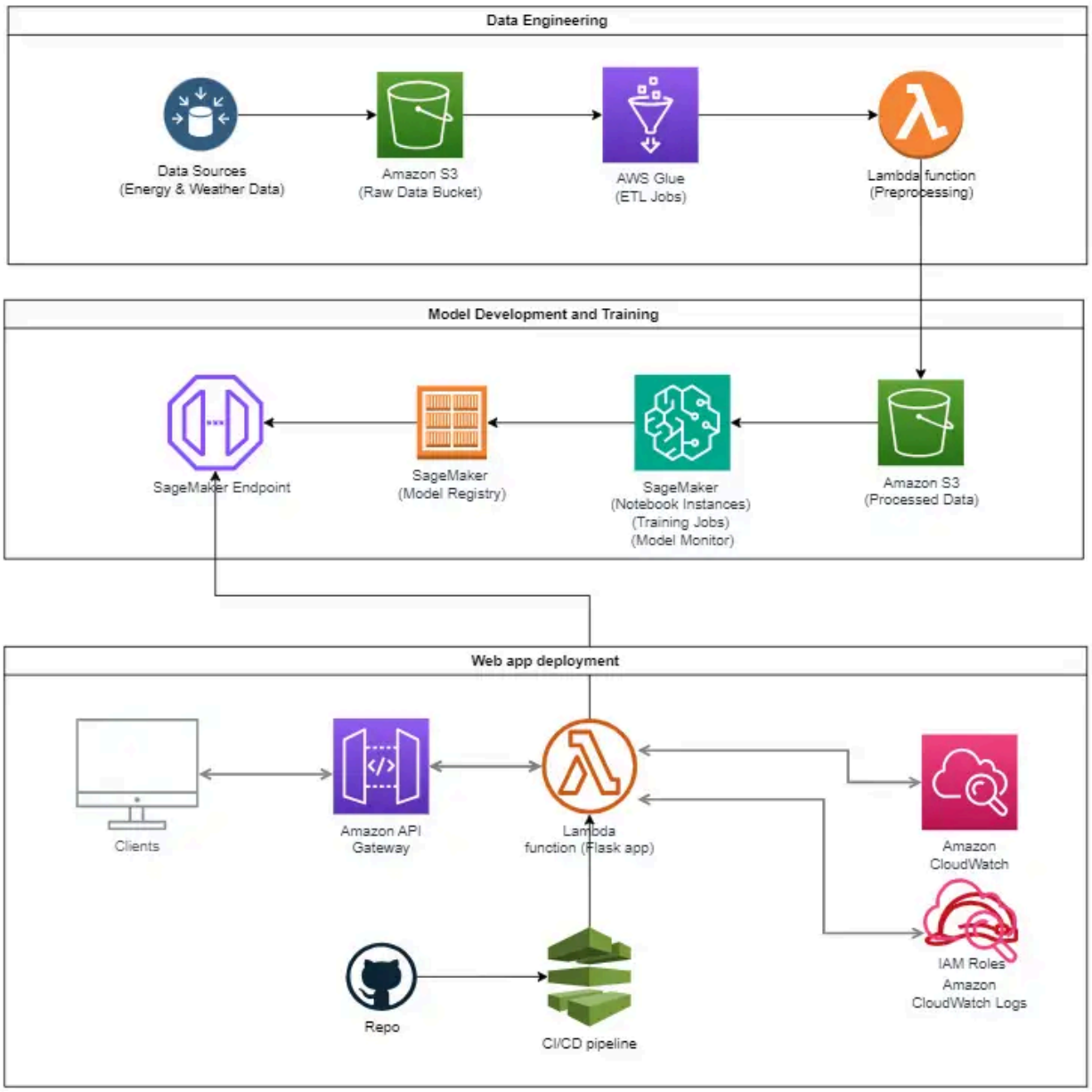
The primary model chosen for this task was the Long Short-Term Memory (LSTM) network. LSTMs are well-suited for time series forecasting because they can capture long-term dependencies and patterns in sequential data.

Train-Test Split: The training and evaluation process utilized TimeSeriesSplit from scikit-learn, which performs cross-validation specifically for time series data.

The dataset was split into 5 folds, ensuring that the training and test sets always maintain their temporal order, preventing data leakage from future to past.

9. Cloud Infrastructure Schema

I have utilized **S3** and **SageMaker** for storing data and training the LSTM mode; and, I recommend a more advanced Cloud Infrastructure Schema for future improvements.



10. Operational Considerations

For scalability, **SageMaker** efficiently handles variable loads by scaling training jobs and inference endpoints according to demand, while **S3** provides unlimited and seamless data storage.

In terms of orchestration, **AWS Glue** manages ETL processes to ensure smooth data processing, and **CodePipeline** automates deployment tasks for consistent updates.

Maintenance is streamlined with regular model updates via **SageMaker Training Jobs** and automated deployment through **CodePipeline**, with **CloudWatch** overseeing system health and log monitoring.

For drift management, **SageMaker Model Monitor** continuously tracks model performance, and periodic retraining through **SageMaker Training Jobs** ensures the model remains accurate and up-to-date.