## 1. What is the problem you want to solve? Why does it interest you?

I plan to predict future energy prices using historical and current data from the U.S. Energy Information Administration (EIA) API. Energy prices fluctuate due to various factors—such as supply and demand, geopolitical events, weather, and seasonal trends—and these fluctuations can significantly impact consumers, businesses, and policymakers. I'm interested in this problem because accurate energy price forecasting can help stakeholders make more informed decisions about budgeting, resource allocation, and strategic planning. Additionally, working on a predictive analytics project in the energy domain will provide valuable experience in handling real-world time-series data and applying data science methods.

## 2. What data are you going to use, and how will you acquire it?

- Data Source: I will primarily use the EIA API, which provides historical and near-real-time data on energy production, consumption, and prices across different regions and energy types (e.g., crude oil, natural gas, electricity).
- Historical energy price data (daily, weekly, or monthly depending on availability).
- Possibly incorporate supplemental data such as economic indicators (e.g., GDP, inflation rates), weather data (for regions with strong seasonal demand), and global commodity market trends.
- Acquisition: I will query the EIA API to gather historical price data for a selected energy product (or multiple products). For external data (e.g., weather, macroeconomic indicators), I may use publicly available APIs or datasets (e.g., NOAA for weather, World Bank or FRED for economic data).

## 3. Brief outline of the approach

- Nature of the problem: This is a supervised learning problem because I have historical price data with corresponding time-stamped features (past prices, production volumes, etc.) and want to predict future prices (a continuous numeric value).
- Specifically, it is a regression problem (predicting a continuous value).
- Methodology:
    1. Data Preprocessing: Clean and aggregate the data, handle missing values, align different time-series frequencies, and engineer relevant features (e.g., lagged prices, rolling averages, seasonal indicators).
    2. Model Selection: Start with traditional time-series forecasting models (e.g., ARIMA, Prophet) and/or regression-based machine learning models (e.g., Random Forest Regressor, Gradient Boosted Trees). If the dataset is large and has complex patterns, consider a deep learning model (e.g., LSTM) for improved performance.
    3. Training & Validation: Split the dataset into training, validation, and test sets. Use time-series cross-validation techniques to avoid data leakage from the future.

4. Hyperparameter Tuning: Use grid search or Bayesian optimization to find optimal model parameters.
5. Evaluation: Compare performance using metrics such as MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and MAPE (Mean Absolute Percentage Error).

- Approach Type: The initial approach will be more of a "traditional" machine learning/time-series forecasting approach (like ARIMA or gradient boosting). If needed, I may experiment with deep learning (LSTMs, RNNs) to capture more complex temporal patterns.

## 4. What will be your final deliverable?

- Application/Service: I will build a small web-based dashboard or an API service where users can input parameters (e.g., region, time horizon) and receive forecasts for energy prices.
- The front end might be a simple dashboard built with a framework like Flask or FastAPI for the API layer, and a lightweight UI (e.g., React or a simple HTML/JavaScript front end).
- The backend will handle data retrieval from the EIA API, model inference, and returning predicted values.

## 5. What computational resources will you need?

- Processing Power (CPU): A standard multi-core CPU (e.g., 4–8 cores) should suffice for initial data cleaning, model training, and inference.
- Memory: At least 8–16 GB of RAM is recommended for handling historical data, feature engineering, and running moderately sized machine learning models. More memory may be required if I incorporate multiple large external datasets or use deep learning models on high-frequency data.
- Specialized Hardware (GPUs):
- For traditional machine learning approaches (ARIMA, XGBoost, etc.), a GPU is not strictly necessary.
- If I decide to train deep learning models (e.g., LSTM or Transformer-based models) on large datasets, a GPU could significantly speed up training. However, for a small to moderate dataset, CPU training might still be feasible.

## Summary

In this project, I will develop a predictive model for energy prices using EIA API data. This supervised regression problem involves collecting and cleaning historical price and production data, engineering time-series features, and training models to forecast future price movements. I plan to deliver a prototype web application or API endpoint that displays forecasts, helping users make informed decisions in energy-related planning and strategy. The required resources primarily include a modest multi-core CPU and sufficient memory, with optional GPU support for more advanced deep learning approaches.