**Workflow for the Project**

Here’s how we’ll tackle it step-by-step:

**Step 1: Define the Data Scope**

* Identify key metrics to scrape (e.g., engine runtime, downtime, fuel consumption trends).
* Focus on scraping ship logs or activity patterns that can be linked to potential maintenance needs.

**Step 2: Scraping Data**

* Use Python libraries like BeautifulSoup or Scrapy to scrape public-facing ship data.
* Leverage APIs (e.g., MarineTraffic API) for structured and real-time ship information.

**Step 3: Data Cleaning**

* Handle missing values, standardize formats, and engineer features from raw scraped data.
* Time-series alignment for sensor data or historical trends.

**Step 4: Analysis and Modeling**

* **EDA**: Identify common trends before equipment failure.
* **Predictive Modeling**: Use machine learning models like Random Forest or Gradient Boosting to predict failures.
* **Anomaly Detection**: Identify unusual operating conditions.

**Step 5: Reporting and Visualization**

* Create detailed dashboards to display insights (e.g., ship health scores, failure probabilities).
* Tools: Python (Matplotlib, Seaborn), or Tableau for interactive dashboards.

**Next Steps for Implementation**

1. **Choose a Target Website:**  
   Would you like to scrape data from **MarineTraffic**, **FleetMon**, or another platform?
2. **Set Up Scraping:**  
   I can guide you in setting up a scraping script for these websites or help you access APIs for structured data.

