**Bike Sharing (Hourly cnt) — Short Analysis & Model Choice**

**Dataset**: UCI Bike Sharing Dataset (hourly file). Target = cnt (total hourly rentals).  
**Goal**: Explore drivers of demand and build a single production-ready model that minimizes **MAE**.

**Executive Summary (why this model?)**

I chose a **HistGradientBoostingRegressor** (sklearn). It balances accuracy, speed, and Operational simplicity.

* Non-linear relationships without manual binning.
* Handles outliers reasonably, insensitive to monotonic scaling.
* Trains fast and supports missing values gracefully.
* Minimal dependency footprint (pure scikit-learn), simplifying maintenance for daily jobs.

**EDA Highlights (business-relevant), Also attached Jupyter Notebbok**

* **Seasonality & weather**: Higher usage in **summer/fall**, depressed by **rain/snow** and extreme temps.
* **Intra-day pattern**: pronounced **commute peaks** (~7–9am, 5–7pm) on **working days**; broader midday peaks on weekends.
* **Calendar effects**: **Working day** and **weekday** interact with time-of-day; **holidays** suppress demand.
* **Trend**: Gradual increase from 2011 to 2012 as service adoption grows.

**Features**

* **Time**: hour-of-day (sin/cos), weekday, month, year, is\_weekend, holiday, workingday.
* **Lags** (leakage-safe): previous **1h**, **2h**, **24h** cnt

**Validation**

* **Split**: chronological; last 20% of rows as **hold-out** validation to mimic future forecasting.
* **Metric**: **MAE** (business-friendly; interpretable in bikes/hour).
* **Baseline**: naive “last-24h same hour” and a linear regression are computed for context.
* **Chosen model**: HistGradientBoostingRegressor with modest hyperparameter search using a **time-series split** (no shuffling).

**MAE** is written to artifacts/metrics.json and printed at the end of training.

**Why one model for production?**

* **Maintainability**: one clean pipeline (src/model\_pipeline.py), deterministic featurization (src/features.py), and a single model artifact (artifacts/model.joblib).
* **Observability**: we log versions, seed, split dates, and metrics to artifacts/metrics.json.
* **Handover-friendly**: small standard library of tools and no exotic dependencies.

**Appendix: Repo Steps**

1. [](http://localhost:63342/markdownPreview/1751956634/markdown-preview-index-236776310.html?_ijt=ba4cnm81g4n6pujfl0ujrrgunf)pip install -r requirements.txt
2. [](http://localhost:63342/markdownPreview/1751956634/markdown-preview-index-236776310.html?_ijt=ba4cnm81g4n6pujfl0ujrrgunf)python -m src.model\_pipeline --data-dir data --artifacts-dir artifacts --seed 42
3. Inspect artifacts/metrics.json and plots under artifacts/plots/.
4. Generate next-day hourly predictions with src/predict\_daily.py.
5. Generate next-day hourly predictions (example date is 2012-12-20; adjust as needed)

python -m src.predict\_daily --model-path artifacts/model.joblib --date 2012-12-20 --data-dir data --out artifacts/preds\_2012-12-20.csv