**Assignment 7.1 — End-to-End Time-Series Ensemble with Exogenous Confidence API**

**Overview**

In this capstone, you’ll finish the pipeline from start to finish and deliver a complete, reproducible project that trains eight base models over the following lookback windows (in days): 365, 270, 180, 90, 60, 30, 14, 1. You will ensemble these base predictions into a meta-model to generate next-step forecasts on your chosen Kaggle time-series dataset. You will also design and integrate a pluggable API for exogenous signals (e.g., news sentiment, macro data) to compute a confidence score/label that accompanies each prediction.

This assignment intentionally builds on:

* Repo & metadata hygiene from Week 1,
* Licensing & citation rigor from Week 2,
* Sequence modeling foundations from Week 3 (attention/BiGRU),
* Multi-lookback training structure & strict saving conventions from Week 4,
* Meta-model ensembling from Week 5, and
* Exogenous/news-sentiment integration patterns from Week 6.

**Learning goals**

* Orchestrate a fully reproducible ML pipeline across multiple lookbacks with strict artifact layouts (Week-4 conventions).
* Train, log, and version base models → meta-model for robust ensembling.
* Implement a clean API contract for exogenous providers and fuse their outputs into a confidence signal.
* Enforce documentation, licensing, and citation best practices across code, data, and models.

**Prerequisites**

* A Kaggle time-series dataset of your choice (document source & license in README and notebook).
* Prior artifacts/concepts from Weeks 1–6 (or equivalent): repo/docs, licensing, model heads/attention (Wk3), multi-lookback training & directory layout, ensembling (Wk5), and API-driven exogenous integration patterns (Wk6).

**Dataset & Modeling Constraints**

* **Dataset**: Must be time-indexed and allow sliding-window feature creation. Cite dataset URL and license.
* **Lookbacks** (all required): **365D, 270D, 180D, 90D, 60D, 30D, 14D, 1D**.
* **1D constraint**: **Do not** use Conv1D on the 1-timestep window; use **BiGRU** (and apply l2 regularization on the last BiGRU where appropriate), consistent with Week-4 guidance.

**Project Structure & Saving Conventions (strict)**

Mirror and extend the Week-4 structure for ensemble inputs. For each lookback LD (e.g., 365D), save artifacts under:

A screen shot of a computer

AI-generated content may be incorrect.

The **root path must end with** .../training/ensemble\_inputs/, and each run must create a **timestamped** folder ending with YYYY-MM-DD\_HH-MM-SS, as in Week-4.

The **meta-model** artifacts must be saved under:

A black screen with white text

AI-generated content may be incorrect.

**Exogenous Confidence API (you will build this)**

Design a **provider interface** so different exogenous sources can be dropped in without changing your ensemble code. For example:

A computer screen with text

AI-generated content may be incorrect.

**Requirements**

* **Security**: Read API keys from **Drive/Env/Secret Manager files**; never hardcode. (Follow Week-6 security hygiene.)
* **Caching**: Implement day-level caching to avoid repeated external calls.
* **Normalization**: Map provider outputs to a consistent schema, e.g., pos/neu/neg + confidence\_label {STRONG, NEUTRAL, WEAK}.
* **Pluggability**: Your pipeline must be callable with **any** provider that implements the contract (e.g., News sentiment like Week-6, macro indicators, alt-data).

**Required Pipeline (End-to-End)**

1. **Repository & Docs (Week-1 style)**
   * Create/update README.md, LICENSE.txt, CITATION.cff at project root with dataset/model citations and license notices.
   * Add a Third-Party Notices section to LICENSE for datasets/models/APIs used.
2. **Data prep & feature windows**
   * Build sliding windows for each required lookback.
   * Split train/val/test; document splits in a top-cell “Project README” inside each notebook.
3. **Train base models for all 8 lookbacks**
   * Architect models appropriately; 1D uses BiGRU (no Conv1D). Apply L2 regularization where appropriate.
   * Save artifacts exactly per the strict layout above.
4. **Generate per-lookback predictions CSVs**
   * Each CSV must align dates and predicted target; plots for loss and predictions are required. (Name patterns as shown.)
5. **Assemble features for meta-model**
   * Load the latest predictions from all eight lookbacks (most recent timestamp subfolders).
   * Construct a feature matrix (columns ordered and saved to feature\_cols.joblib).
6. **Train the meta-model**
   * **Baseline**: Ridge regression (consistent with Weeks 4–6); you may compare against another linear model.
   * Save meta\_model.joblib and feature\_cols.joblib.
7. **Integrate the Exogenous Provider** (confidence)
   * Implement provider (e.g., NewsAPI + FinBERT pattern from Week-6) and produce normalized sentiment metrics + label.
   * Fuse the meta-model prediction with the confidence label in a single summary.
8. **Prediction & Logging**
   * Produce a prediction summary for the next step on your dataset:
     + Predicted value, previous actual, predicted % change
     + Confidence label and exogenous metric averages
   * Append an entry to ensemble\_prediction\_log.csv with: timestamp, predicted value, (actual if available), % change, exogenous scores/label, and the base-model inputs used. (Week-6 style logging.)
9. **Reproducibility & Config**
   * Centralize paths, lookbacks, and seeds.

**Deliverables**

1. **Code & Notebooks**
   * One training notebook per lookback (8 total), each producing required artifacts and following naming/layout rules.
   * One meta-model training notebook/script.
   * One end-to-end runner that: loads latest lookback predictions → applies meta-model → queries exogenous provider → prints & logs summary.
2. **Artifacts**
   * Eight timestamped lookback folders with \*\_predictions.csv, \*\_predictions.png, \*\_loss.png.
   * training/meta\_model/ with meta\_model.joblib, feature\_cols.joblib, ensemble\_prediction\_log.csv.
3. **Documentation**
   * Root README.md with project overview, dataset/model sources & licenses, run instructions, and directory map (Week-1 style).
   * LICENSE.txt with chosen license and third-party notices.
   * CITATION.cff with authors/project/date/license and references (dataset, models, APIs).
4. **Example Output**
   * Screenshot (or cell output) of a final prediction summary (prediction + % change + confidence + exogenous averages).

**Acceptance checklist**

* All eight lookbacks trained; artifacts saved under the strict Week-4 directory layout & timestamping.
* 1D model uses BiGRU (no Conv1D) with appropriate L2 on final BiGRU when needed.
* meta\_model.joblib and feature\_cols.joblib saved; features aligned exactly.
* Exogenous provider interface implemented; keys not hardcoded; caching enabled.
* Final prediction summary includes confidence label and normalized exogenous metrics.
* ensemble\_prediction\_log.csv updated with all required fields.
* README.md / LICENSE.txt / CITATION.cff completed with correct citations & licenses

Grading rubric (100 pts)

| Criterion | Pts | Description |
| --- | --- | --- |
| Correctness & Reproducibility | 25 | End-to-end run succeeds; strict directory/timestamping; deterministic config; clear runner. |
| Base Models (8 Lookbacks) | 20 | Architectures appropriate per window; 1D uses BiGRU + L2 when appropriate; clean plots & CSVs. |
| Meta-Model Ensembling | 15 | Proper feature assembly & ordering; strong baseline Ridge; artifacts saved & documented. |
| Exogenous Confidence API | 15 | Clean provider interface; secure key handling & caching; normalized scores + confidence label; integrated into summary. |
| Logging & Evaluation | 10 | Accurate summary metrics; comprehensive ensemble\_prediction\_log.csv with inputs & exogenous fields. |
| Docs, Licensing, Citation | 10 | README, LICENSE (with notices), CITATION.cff complete and accurate. |
| Code Quality | 5 | Parameterized, modular, DRY; clear comments & function boundaries. |
| Professionalism | 5 | Clear instructions to run; tidy repo; no secrets in code or outputs. |

**Implementation hints**

* Start by finishing the **longest lookbacks first** (365→270→180…) to stabilize shapes & training, then replicate patterns across the rest (as recommended in Week-4).
* Keep your **feature column order** fixed and versioned (save to feature\_cols.joblib), then always align live predictions to this order (Week-6 pattern).
* Your **confidence mapping** can be threshold-based on normalized exogenous scores (e.g., pos - neg > τ ⇒ STRONG), but document the heuristic and make it configurable.