**Assignment 5.1 — Meta-Model Ensembling from Week-3 Lookbacks**

**Objective**

Build a reproducible Week-5 meta-model that ingests the *\_predictions.csv* files produced in Assignment 4.1 (Week-4), aligns them by date, and fits an ensemble regressor (meta-model) in anticipation of predicting next-day closing price in Week 6. Base your solution on the shared example notebook Nvidia\_Next\_Day\_Closing\_Meta\_Model\_Train\_Week5.ipynb and generalize it to support different meta-model choices.

**Inputs & Assumptions (must match your Week-4 outputs)**

* **Required lookbacks**: (e.g. ["365D", "90D", "14D", "1D"]) (you may include additional lookbacks if you trained them).
* **Folder layout (inputs)** — one timestamped run per lookback:

A screenshot of a computer

AI-generated content may be incorrect.

* Note: The starter notebook defaults to "/content/drive/My Drive/Nvidia\_Stock\_Market\_History/Training/ensemble\_inputs". If your Week-4 path used lowercase training, update the constant accordingly.
* **CSV schema (per lookback)**:
  + Columns include Date, Predicted\_Close, and Actual\_Close (from your Week-4 pipeline).
  + File name should contain a lookback token (e.g., ...365D\_...\_predictions.csv) so the notebook can infer the lookback. If your Week-4 file names lack this, extract the lookback from the parent folder name instead.

**Meta-Model Constraints**

* **Default**: Ridge(alpha=1.0)
* **Supported choices (parameterized)**: LinearRegression, Ridge, Lasso, ElasticNet.  
  (You may add others like RandomForestRegressor or XGBRegressor but ensure feature importance reporting still works.)
* **Feature set**: strictly the Predicted\_Close columns from each lookback (renamed to Pred\_<LOOKBACK>), aligned by Date.
* **Target**: Actual\_Close.

**Required Pipeline (generalize the starter notebook)**

1. **Setup / Mount**
   * Mount Google Drive (if using Colab).
   * Define constants:
     + ENSEMBLE\_INPUTS\_FOLDER (defaults to the path above).
     + LOOKBACK\_FOLDERS = ["365D", "90D", "14D", "1D"] (extend if you have more).
     + META\_MODEL\_SAVE\_FOLDER (see “Outputs & Saving” below).
2. **Discover latest run per lookback**
   * For each lookback folder, select the latest timestamped subfolder (sorted by YYYY-MM-DD\_HH-MM-SS suffix).
   * From that subfolder, load the first \*\_predictions.csv.
3. **Load & align**
   * For each CSV:
     + Keep Date, Predicted\_Close.
     + Rename to Pred\_<LOOKBACK>.
   * Inner-join all lookbacks on Date.
   * Merge in Actual\_Close from any CSV (they should match).
4. **Train**
   * Train the chosen meta-model on X\_train = [Pred\_365D, Pred\_90D, Pred\_14D, Pred\_1D, ...], y\_train = Actual\_Close.
   * If using regularized models, allow a simple hyperparameter search (e.g., a small grid over alpha for Ridge/Lasso; alpha and l1\_ratio for ElasticNet). Log the chosen params.
5. **Evaluate**
   * Report metrics on train and test (at minimum): R², MAE, RMSE.
6. **Save artifacts**
   * Model serialized with joblib.
   * Feature column order (feature\_cols.joblib) to preserve lookback ordering at inference.
   * Save to similar directory endpoint: "/content/drive/My Drive/Nvidia\_Stock\_Market\_History/Training/Meta\_Model\_Trained".
   * Use similar naming conventions:
     + features\_cols.joblib
     + meta\_model\_ridge.joblib

**Notebook Requirements**

* **One notebook total (Week-5)** that:
  + Is cleanly **parameterized** (meta-model choice, alpha grid, paths, lookbacks).
  + Uses functions to avoid duplication (discovery → load/align → split → train → eval → save).
  + All code is commented.

**Deliverables**

* Colab notebook (e.g., Assignment\_4\_1\_Meta\_Model\_Ensemble.ipynb) that:
  + Discovers & loads the latest \*\_predictions.csv per required lookback.
  + Trains your chosen meta-model.
  + Saves all artifacts to the timestamped output folder (structure above).

**Grading Rubric (100 pts)**

* **Correctness & Reproducibility (40 pts)**
  + Exact input discovery (latest timestamp per lookback) and strict output structure with timestamped folder.
  + End-to-end run produces all artifacts deterministically.
* **Model Design (30 pts)**
  + Proper meta-model setup; regularization where applicable.
  + Clear, justified hyperparameter selection (even a small grid).
* **Code Quality (30 pts)**
  + Parameterized, modular functions; minimal duplication; clear comments.
  + Robustness to missing files / mismatched dates (clean error messages).

**Starter Constants (mirror the uploaded notebook; adjust if your Week-4 path differs)**

* ENSEMBLE\_INPUTS\_FOLDER = "/content/drive/My Drive/Nvidia\_Stock\_Market\_History/Training/ensemble\_inputs"
* LOOKBACK\_FOLDERS = ["365D", "90D", "14D", "1D"]
* META\_MODEL\_SAVE\_FOLDER = "/content/drive/My Drive/Nvidia\_Stock\_Market\_History/Training/Meta\_Model\_Trained"  
  → Create a timestamped subfolder inside this at runtime.

**Notes & Hints**

* If your Week-4 CSV filenames don’t include a XXXD token, infer the lookback from the parent folder instead of the filename.
* If you add tree-based meta-models, keep the linear model as a baseline and compare metrics.
* Be mindful of date alignment: use inner joins and print the merged row count.