**Week 6 – Instructor Lecture Notes: Meta‑Model Next‑Day Nvidia Close (with News Sentiment API)**

**Orientation & Architecture**

* **Inputs:** Latest per‑lookback base‑model predictions (1D, 14D, 30D, 60D, 90D, 180D, 270D, 365D) + 5‑day news sentiment (FinBERT on NewsAPI articles).
* **Model:** Pre‑trained Ridge meta‑model that ingests the base predictions (and engineered features) to output next‑day closing price.
* **Runtime:** Google Colab + Drive for persistent artifacts, PyTorch (for FinBERT), Transformers, and joblib for model I/O.
* **Outputs:** Printed summary and an append‑only CSV log with timestamp, prediction, deltas, and sentiment aggregates.

**Pro tip to students:** This notebook is the inference & orchestration layer for a two‑stage system: (1) base forecasters (trained earlier), (2) a light meta‑learner that ensembles their signals and conditions on news sentiment.

**Load Dependencies into the Colab Runtime (pip installs)**

**What it does**  
Installs ipywidgets, pandas 2.2.2, tqdm, PyTorch (CUDA 12.4 build), newsapi-python, joblib, and requests. Transformers are implied; FinBERT is loaded later.

**Why it matters**

* Version pinning reduces breakage across Colab sessions.
* CUDA build selection (e.g., +cu124) must match the notebook runtime; otherwise GPU acceleration for FinBERT will fail (falls back to CPU).

**SME cues**

* Call out binary‑compatibility issues (PyTorch ↔ CUDA ↔ Driver).
* Emphasize that pip installs inside notebooks are environment‑scoped and ephemeral; containerize in production.

**Imports & Core Modules**

**What it does**  
Imports os, glob, pandas, joblib, datetime, newsapi, requests, Transformers (AutoTokenizer, AutoModelForSequenceClassification), PyTorch, and json.

**Why it matters**

* Separates dataI/O (pandas, json), model I/O (joblib), NLP (Transformers + Torch), and HTTP (requests) concerns.

**SME cues**

* Tie imports to layered architecture: data plumbing → model plumbing → external services.

**Mount Google Drive**

**What it does**  
Mounts Drive at /content/drive.

**Why it matters**

* Provides statefulness across ephemeral sessions (models, caches, logs).

**SME cues**

* Contrast with GCS/S3 in production.
* Mention idempotency: re‑mounts are safe, but credentials are interactive.

**Read & Configure News API Settings**

**What it does**  
Reads a NewsAPIkey from Drive, creates NewsApiClient, and sets cache paths (NEWS\_CACHE\_DIR, NEWS\_CACHE\_PATH).

**Why it matters**

* Avoids hard‑coding secrets in source.
* Prepares a local cache to respect API quotas and improve latency.

**SME cues**

* For prod, use a secret manager and env vars; rotate keys.
* Define a rate‑limit policy and retry/backoff strategy around NewsAPI calls.

**Load FinBERT Model & Tokenizer**

**What it does**  
Loads ProsusAI/FinBERT tokenizer and sequence‑classification model.

**Why it matters**

* Provides finance‑domain sentiment (pos/neu/neg logits) used to grade ensemble confidence.

**SME cues**

* Switch model to eval() and use torch.no\_grad() for inference‑only path; move to GPU if available (.to('cuda')).
* Discuss **token truncation** (truncation=True) and its effect on long articles.

**Meta‑Model Prediction Configuration**

**What it does**  
Declares META\_MODEL\_PATH (Ridge .joblib) and LOG\_PATH for an append‑only CSV log.

**Why it matters**

* Externalizes paths for artifact portability and auditability (logs).
* Sets up a model registry style workflow (lightweight).

**SME cues**

* In prod, these live behind a model registry (e.g., MLflow) with versions and signatures.

**Discover Latest Base‑Model Prediction Files**

**What it does**

* Defines the ordered list of lookbacks: ["365D","270D","180D","90D","60D","30D","14D","1D"].
* Scans ensemble\_inputs/<LOOKBACK> subfolders, sorts by encodedtimestamp, and picks the latest \*\_predictions.csv per lookback.

**Why it matters**

* Guarantees the meta‑model consumes synchronized, most recent signals from each base learner.

**SME cues**

* Explain the timestamp parsing helper (e.g., extract\_dt) and importance of a filename convention.
* Ensure the sort key is robust (don’t parse arbitrary text; validate schema).

**Prepare Latest Predictions for Meta‑Model**

**What it does**

* Loads each latest \*\_predictions.csv, takes the last row, and extracts Predicted\_Close (and reads Actual\_Close for reporting).
* Constructs a one‑row X\_input with columns like Pred\_30, Pred\_60, … (one per lookback).
* Loads a saved feature\_cols.joblib and reindexes X\_input to match the training‑time column order.

**Why it matters**

* Prevents **feature drift**: meta‑model sees identical feature ordering at train and serve time.

**SME cues**

* Emphasize contract tests between training and inference: same columns, dtypes, scales.

**Fetch Latest Nvidia News Articles (past 5 days)**

**What it does**  
Defines fetch\_today\_articles() that queries NewsAPI (English, “Nvidia”, last 5 days, sorted newest) and returns a simplified list of {date, title, description}.

**Why it matters**

* External exogenous variable capturing market sentiment that base models don’t see.

**SME cues**

* Discuss API parameters (query, date window, page\_size/pagination) and deduplication (many outlets syndicate the same wire story).

**Cache & Load Recent News Articles**

**What it does**

* Ensures NEWS\_CACHE\_DIR exists.
* Loads/initializes news\_cache.json keyed by date.
* If today is not in cache, fetches today’s list and trims cache to last 5 days, then writes it back.
* Flattens all kept days into all\_articles.

**Why it matters**

* Reduces API calls, stabilizes latency, and ensures a consistent sentiment window for the ensemble.

**SME cues**

* Mention cache eviction policy and its impact on reproducibility.
* Highlight time zone alignment: NewsAPI returns UTC‐ish timestamps; ensure cache dates are comparable.

**Compute FinBERT Sentiment over Cached Articles**

**What it does**

* Iterates all\_articles, concatenates title + description, tokenizes with FinBERT, runs the model to produce logits, applies softmax to get pos/neu/neg probabilities, and averages across articles.
* Maps average scores to a coarse confidence label: STRONG (pos ≥ 0.5), WEAK (neg ≥ 0.5), else NEUTRAL.

**Why it matters**

* Converts qualitative news into numerical features that modulate how much we trust the ensemble’s direction.

**SME cues**

* Use model.eval() and torch.no\_grad().
* Consider weighted averages (by outlet, recency) and robust stats (median, winsorization) to resist outliers.

**Load Ridge Meta‑Model & Predict**

**What it does**  
Loads meta\_model\_ridge.joblib and calls .predict(X\_input) to produce ensemble\_pred (float close price).

**Why it matters**

* Stacked generalization: the Ridge meta‑model blends base predictors, often improving bias–variance trade‑off.

**SME cues**

* Explain whyRidge: linear, well‑behaved with correlated inputs, L2 regularization guards against co‑linearity among base predictions.

**Display Ensemble Prediction Summary**

**What it does**  
Computes delta\_pct = (ensemble\_pred - actual\_close) / actual\_close \* 100 and prints a human‑readable summary including the sentiment aggregates and qualitative confidence.

**Why it matters**

* Presents a single source of truth for the day’s call, with interpretable context.

**SME cues**

* Remind students that directional accuracy can be assessed even without tomorrow’s truth yet.
* In later work, compare predictions to realized close for online monitoring.

**Log Ensemble Prediction with Metadata (CSV)**

**What it does**

* Assembles a one‑row DataFrame with date, timestamp, ensemble\_pred, actual\_close, delta\_pct, confidence, and averaged sentiment probabilities.
* Ensures log directory exists, creates the log if missing, and otherwise appends with column order preserved.

**Why it matters**

* A lightweight MLOps audit trail: who/when/what was predicted, with exogenouscontext baked in.

**SME cues**

* Logging is the first step toward drift detection and post‑hoc analysis.
* In prod, ship logs to a time‑series store or object storage with partitioning by date.

**Cross‑Cutting Concerns & Best Practices (discuss after the walkthrough)**

**Reproducibility**

* Set global random seeds, capture package versions, and snapshot artifacts (meta‑model, feature column order).
* Ensure feature contracts are enforced (unit tests: column names, order, dtype).

**Performance**

* Batch FinBERT inference for many articles; pre‑tokenize and cache token IDs.
* Avoid CPU/GPU thrash (don’t move tensors between devices repeatedly).

**Security**

* Never commit API keys; load via secret managers.
* Sanitize and log only non‑PII metadata; NewsAPI content may include sensitive entities.

**Reliability**

* Wrap API calls with retry/backoff; log failures and degrade gracefully (use cached sentiment).
* Validate file discovery across lookbacks; fail with actionable errors.

**Evaluation & Monitoring**

* Track prediction vs realized close with lag, compute MAE/RMSE, directional hit‑rate, and calibration (is delta% predictive?).
* Monitor concept drift (feature distributions, residuals) and data freshness.

**Extensibility**

* Replace Ridge with stacking (e.g., Lasso, ElasticNet, Gradient Boosting) and compare via cross‑validation.
* Enrich features: volatility indices, options flow, macro calendars, or source‑weighted sentiment.