**Assignment 6.1 — Integrating News Sentiment into the Meta-Model Pipeline**

**Objective**

You will extend your existing meta-model pipeline from Week 5 to integrate recent news sentiment as an additional context signal for predicting Nvidia’s next-day closing price. This assignment builds on the meta-model ensembling work you completed in Assignment 5.1 - Meta-Model Ensembling from Week-4 Lookbacks, reusing the serialized model and feature ordering from that week. You will follow the structure and methodology demonstrated in the Week 5 in class lab.

**Prerequisites**

* Week 5 artifacts:
  + meta\_model\_ridge.joblib — trained Ridge meta-model
  + feature\_cols.joblib — exact feature column ordering from training
  + Both located in:



* Customize to your path used in Assignment 5.1
* Access to the Week-6 notebook:
  + Nvidia\_Next\_Day\_Closing\_Meta\_Model\_Prediction\_w\_API\_Call\_Week6.ipynb
* Familiarity with mounting Google Drive, importing libraries, and setting up the Colab environment (review earlier lessons if needed).

**High-Level Steps**

1. **Review Week-6 notebook structure**
   * Understand where the notebook loads base-model predictions, applies the feature order, and uses the meta-model to produce the ensemble prediction.
   * Identify where the NewsAPI call, FinBERT sentiment calculation, and prediction logging occur.
2. **Set up a NewsAPI account**
   * Register at <https://newsapi.org/register>.
   * Create and save in a .txt file an API key for NewsAPI services.
   * Review the Python client library docs at <https://newsapi.org/docs/client-libraries/python>.
3. **Securely store your NewsAPI key**
   * Save your API key in a .txt file inside Google Drive (e.g.,  
     /content/drive/My Drive/Nvidia\_Stock\_Market\_History/API\_keys/newsapi\_key.txt).
   * Ensure the notebook reads this key from the file.
   * NEVER HARDCODE AN API KEY!
4. **Integrate the NewsAPI client into the pipeline**
   * Use the Week-5 notebook example to instantiate NewsApiClient with your key.
   * Configure the API call to fetch articles about “Nvidia” or “NVDA” from the past 5 days.
     + Customize the HTTP request to return news articles about the financial topic covering your chosen Kaggle dataset.
   * Implement caching so the same articles aren’t repeatedly fetched within a day.
5. **Compute sentiment scores with FinBERT**
   * Pass each news article’s title and description to FinBERT.
   * **Calculate average positive, neutral, and negative scores** across all fetched articles.
   * Derive a qualitative confidence label (e.g., STRONG, NEUTRAL, WEAK) based on these averages.
6. **Load the Week-5 meta-model and latest lookback predictions**
   * From the ensemble input directories (1D, 14D, 30D, etc.), identify and load the most recent predictions.
   * Align these values to the saved feature\_cols.joblib ordering.
   * Pass them to the loaded meta-model to generate the next-day predicted close.
7. **Combine prediction and sentiment output**
   * Display a summary including:
     + Predicted close
     + Previous close
     + Predicted percentage change
     + Confidence label
     + Average sentiment scores
8. **Log results for reproducibility**
   * Append a new row to ensemble\_prediction\_log.csv in the Week-5 output directory: (e.g., “content/drive/My Drive/Nvidia\_Stock\_Market\_History/Training/Meta\_Model\_Trained/…”)
   * Ensure the log contains:
     + Date/time of prediction
     + Predicted close, actual close, percentage change
     + Confidence label and sentiment averages
     + Base-model prediction inputs

**Deliverables**

1. Updated notebook implementing the above steps (no API key in the code or output).
2. Example prediction summary (output cell screenshot or text).
3. Updated log file (ensemble\_prediction\_log.csv) in the Week-5 directory.

**Grading Rubic**

| **Criteria** | **Description** | **Weight** |
| --- | --- | --- |
| **Security Hygiene** | **API key is stored securely in a Drive text file (or secret/env variable) and never exposed in code or output.** | **20%** |
| **Integration Accuracy** | **Correctly uses Week 5 artifacts (meta\_model\_ridge.joblib, feature\_cols.joblib) and latest lookback predictions in the Week 6 pipeline.** | **30%** |
| **Sentiment Processing** | **Fetches 5 days of Nvidia news, applies FinBERT correctly, and computes accurate average positive, neutral, and negative sentiment scores.** | **20%** |
| **Prediction Logic** | **Produces accurate next-day close prediction using the meta-model, displays correct summary statistics, and applies confidence labeling.** | **15%** |
| **Reproducibility & Logging** | **Appends a complete, well-structured row to ensemble\_prediction\_log.csv including sentiment, prediction, and input features.** | **15%** |