**Advanced AI Forecasting with TensorFlow and Natural Language Processing**

**Week 2 — Time Series Foundations, Dataset Exploration, and Lookback Theory**

**Note:** This Colab Notebook can be fully executed on a CPU runtime. A GPU is not required for the setup phase, and all steps in this section will run without hardware acceleration.

**Learning Objectives**

By the end of Week 2, students should be able to:

* Understand the fundamentals of time series forecasting.
* Navigate and explore the NVIDIA stock dataset.
* Perform exploratory data analysis (EDA) and basic feature engineering.
* Understand the concept of lookback windows in sequential models.
* Discuss ethical considerations, licensing, and compliance.

**How this supports the Week 2 pipeline**

For Weeks 2–6, all students will work in Google Colab, which provides free-tier access to T4 GPU. Also, if students prefer access to the A100 or L4 GPU with priority usage, they can sign up for “No Cost for students and educators” subscription at: <https://colab.research.google.com/signup>. Remind students to use their UAT student email when signing up. Finally, students can also pay a one-time payment of $9.99 for 100 Compute Units (<https://colab.research.google.com/signup>) for access to the A100 or L4 GPUs, which is more than enough to run all the notebooks for the entire class. This ensures that every student, regardless of local hardware, can:

* Experiment with sequential models in real time.
* Compare results across lookback configurations.
* Begin thinking about how the same pipelines will later run unattended in a local or server-based automated environment.

**Common pitfalls to emphasize**

* **Driver-toolkit mismatch:** Ensure Windows NVIDIA driver is up to date before installing framework wheels.
  + If using Colab, remind students that the runtime already includes CUDA/cuDNN.
* **Python/NumPy/TensorFlow ABI mismatches:** Pin versions when needed (as shown in the lab script) to avoid binary incompatibilities.
* **Dataset formatting:** Enforce *no* $ in prices and *no* commas in volumes if students manually append rows.

**Lecture Topics (with Environment Tie-ins)**

1. **Introduction to Time Series Forecasting — Key Points**

* **Definition:** *Time series data consists of observations recorded in chronological order, where the sequence matters.*
* **Trend:** *The long-term upward or downward movement in the data over time (e.g., steady stock price growth).*
* **Seasonality:** *Regular, repeating patterns tied to time intervals (e.g., monthly sales peaks, quarterly cycles).*
* **Noise:** *Random, unpredictable fluctuations not explained by trend or seasonality (e.g., sudden market shocks).*
* **Why sequence models?**
  + *They process data in order, retaining memory of previous steps.*
  + *Models like LSTMs/GRUs capture temporal dependencies—how past values influence future predictions.*
  + *They are well-suited for patterns where context over time impacts the outcome.*

**2. Dataset Exploration (NVIDIA Stock)**

* EDA: Examine structure, column types, and descriptive statistics.
* Visualizations: Plot closing prices, volume, and trends to spot patterns.
* Missing values: Identify gaps or anomalies; decide whether to impute or drop them.
* Rolling means: Smooth short-term fluctuations to reveal underlying trends.
* Formatting: Ensure no $ in prices and no commas in volumes to maintain CSV compatibility.

**3. Feature Engineering (Hands-On)**

* Moving averages: Capture short-, medium-, and long-term trends (e.g., MA20, MA50, MA200).
* Volatility: Rolling standard deviation to measure variability.
* Returns: Percentage change over various intervals to measure momentum.
* Bollinger Bands: Upper/lower bounds for price relative to volatility.
* Clean dataset: Remove NaNs from rolling calculations and save for modeling.

**4. Lookback Windows (Theory)**

* Definition: Number of past timesteps used to predict the next value.
* Purpose: Allows LSTM/GRU models to learn temporal dependencies from recent history.
* Trade-offs:
  + Short lookback (e.g., 1–14 days): Faster training, captures immediate trends but may miss long-term patterns.
  + Long lookback (e.g., 180–365 days): Captures broader trends but increases model complexity and risk of overfitting.

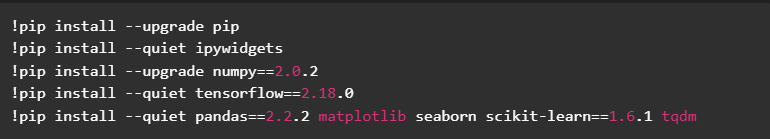
**Materials Needed**

* Week 2 Colab Notebook (to be distributed).
* Optional: Slides on time series forecasting concepts.
* License & citation table (for reference and discussion).
* WSL2 + CUDA installation deck for Windows users (reference for local GPU workflows).

**Lab Activities (NSMH\_EDA\_Feature\_Engineering\_Week1.ipynb using Google Colab) — Code Walkthrough & Rationale**

Below is an instructor-facing explanation of each major code block from the Week 2 lab script. Use this to guide live walkthroughs and anticipate student questions.

1. **Environment & Package Setup**

****

**Why:** Ensures a consistent, known-good set of binary wheels for TensorFlow and dependencies. Pinning versions reduces Application Binary Interface conflicts (e.g., NumPy ↔ TF) and makes results reproducible across students. ipywidgets improves interactivity in Colab; scikit-learn provides scalers.

1. **Library and Modules Imports and Version Diagnostics**

A screen shot of a computer

AI-generated content may be incorrect.

**Why:** Quick smoke test to ensure that expected versions are active. Early failure here saves time by surfacing environment issues before long training jobs.

1. **Mount Google Drive (Colab) & Create Project Directory**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Why:** Persist datasets, models, and plots across sessions. Encourages good project hygiene (clear base paths and artifact directories), which is essential for automated pipelines.

1. **Dataset Upload & Canonical Pathing**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

**Why:** Normalizes the dataset location under the project directory. This also guarantees consistent paths for subsequent cells (important when scripting automation). Reinforce **no `$` in prices and no commas in volumes** when students create/append data manually.

1. **Verifying Project Directory Contents Before Pipeline Execution**

****

**Why:** Lists the contents of the target Google Drive project directory in a readable format, allowing quick verification that required datasets and files are present before running preprocessing or training steps—essential for preventing file path errors and ensuring the pipeline starts with the correct inputs.

1. **Base Directories & User Inputs**

**A computer screen with white and green text

AI-generated content may be incorrect.**

**Why:** Parameterizes runs for automation (unique timestamped folders). lookback controls sequence length for LSTM/GRU inputs; naming ensures plots/weights don’t overwrite between experiments.

1. **Reproducibility Settings**

**A black screen with white text

AI-generated content may be incorrect.**

**Why:** Makes data shuffles, weight initializations, and some CUDA kernels deterministic, increasing repeatability of training/validation results—a must for fair comparisons across lookbacks and features.

1. **Load, Sort, and Feature EngineerA screen shot of a computer program

   AI-generated content may be incorrect.**

**Why:**

* Sorting by time preserves causal order (no target leakage).
* Rolling statistics & returns encode trend/volatility/momentum regimes the model can learn.
* injects an immediate autoregressive signal; adds relative positioning within a local window.
* `dropna()`is required because rolling windows introduce missing values at sequence starts.
  + When you use methods like:
    - A black background with white text

      AI-generated content may be incorrect.
    - the first 19 rows will have NaN values, because there aren’t enough prior data points to calculate a 20-period rolling mean yet.

**8) Feature Scaling (Per-Feature MinMax)**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Why:** LSTMs/GRUs are sensitive to feature scale; per-feature scaling stabilizes training and speeds convergence. Storing each scaler enables inverse transforms later (e.g., mapping predictions back to price scale).

**9) Create Supervised Sequences (Lookback → (X, y))**

**A computer screen shot of white text

AI-generated content may be incorrect.**

**Why:** Transforms a univariate/ multivariate time series into supervised learning format for sequence models. Each training example is a **window** of lookback timesteps; the target is the next-step close. The first column in scaled\_data is Close, so y aligns with the prediction target.

**10) Temporal Train/Val/Test Split**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

**Why:** Preserves time order (no shuffling) to mimic real-world forecasting, where future data isn’t visible during training. The 70/20/10 split supports hyperparameter tuning on validation and truly unseen evaluation on the test set.

**Instructor Tip:** Emphasize that *random shuffles break temporal causality*. If students need balanced splits, use contiguous blocks or rolling-origin evaluation.

**Ethics & Compliance**

| **Component** | **Source** | **URL** | **License** | **License URL** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| **NVIDIA Stock Dataset** | **Kaggle - Adil Shamim** | [**Link**](https://www.kaggle.com/datasets/adilshamim8/nvidia-stock-market-history) | **CC0 1.0 Universal** | [**Link**](https://creativecommons.org/publicdomain/zero/1.0/) | **Used for time series forecasting. The dataset is in the public domain and free to use.** |
| **FinBERT** | **ProsusAI** | [**Link**](https://huggingface.co/ProsusAI/finbert) | **Apache License 2.0** | [**Link**](https://www.apache.org/licenses/LICENSE-2.0) | **Not an official Prosus product. Result of an intern research project. Contact: Dogu Araci (**[**dogu.araci@prosus.com**](mailto:dogu.araci@prosus.com)**), Zulkuf Genc (**[**zulkuf.genc@prosus.com**](mailto:zulkuf.genc@prosus.com)**). Repo:** [**GitHub**](https://github.com/ProsusAI/finBERT) |

* Model/source transparency and citation continue to apply.
* Preview Kaggle dataset (CC0 1.0) and FinBERT license (Apache 2.0) and dataset reuse rights.

**Instructor Prep Notes**

* Ensure students can run the Colab notebook or Jupyter locally.
* Provide visual aids for lookback sequence creation.
* Allocate time for Q&A on feature scaling, temporal splits, and time series assumptions.
* Encourage students to keep artifacts (scalers, plots, metrics) under timestamped folders for repeatable automation.

**Key Takeaways to Reinforce**

* Feature engineering + scaling + supervised sequence construction are the core pre-processing steps for TensorFlow Sequential models.
* Temporal splits and deterministic seeds are critical for fair validation.