Task 5 Report – Implementation of Advanced Stock Prediction Functions

1. Summary of Implementation Effort

This task extended the original single-step, univariate stock prediction model to cover more advanced forecasting:

- **Multistep Prediction** – Predict multiple future days (e.g. next 5 days) instead of only tomorrow's price.
- **Multivariate Prediction** – Use multiple input features (Open, High, Low, Close, Volume) instead of just closing price.
- **Combined Approach** – Integrate multistep and multivariate capabilities for a more comprehensive forecasting model.

---

2. Analysis of Less Straightforward Code Lines

2.1 Dynamic Sequence Generation

```
for i in range(prediction_days, len(price_data) - future_steps + 1):
    x_data.append(price_data[i-prediction_days:i])
    y_data.append(price_data[i:i+future_steps])
```

- **Why the - future_steps + 1?**
  This ensures you don't try to index beyond the end of the array when forming y_data sequences of length future_steps. The +1 accounts for inclusive slicing bounds.
- **Reference Link:** TensorFlow's tutorial on time series forecasting (which covers single-step and multistep forecasting) — Time Series Forecasting – TensorFlow TensorFlow

---

2.2 Multivariate Feature Extraction

```
for j in range(i-prediction_days, i):
    day_features = [scaled_data[col].iloc[j] for col in feature_columns]
    x_sequence.append(day_features)
```

- Here we convert each day into a vector containing all features (e.g. Open, High, Low, Close, Volume) rather than a single price. This is required so LSTM sees all features each time step.
- **Reference Link:** The Keras LSTM documentation talks about input shape as (batch_size, timesteps, features) which supports this structure. keras.io

---

2.3 Scaler Dictionary Management

```
for col in feature_columns:
    scaler = MinMaxScaler(feature_range=(0, 1))
    scaled_data[col] = scaler.fit_transform(data[col].values.reshape(-1, 1))
    scalers[col] = scaler
```

- Because each feature may have a different range (e.g. Volume vs Price), we fit a separate scaler for each, and store them so we can invert scaling later.
- **Reference Link:** scikit-learn's MinMaxScaler documentation explains scaling of individual features. scikit-learn.org
- Also, the general scikit-learn preprocessing guide covers scaling methods. scikit-learn.org

---

2.4 Model Output Layer Configuration

model.add(Dense(future_steps, activation="linear"))

- In multistep prediction, the model's final Dense layer must output *future_steps* values, producing a vector instead of a single scalar.
- This aligns with sequence-to-sequence or multi-output regression design in neural networks.

2.5 Array Reshaping for LSTM Input

x_data = np.reshape(x_data, (x_data.shape[0], x_data.shape[1], 1))

- LSTM layers require 3D input: (samples, timesteps, features).
- In the univariate case, features = 1, hence the reshape.
- The Keras RNN/LSTM documentation outlines this expected input shape. keras.io

---

3. Experimental Results Summary

3.1 Multistep Prediction Experiments

- **Configuration:** Input = 60 days of closing prices; Output = 5 future days.
- **Results:**
    - Shapes: (samples, 60, 1) → (samples, 5)
    - Model converged during training, but validation error increases for longer prediction horizons.
    - Days 1–2 predictions are better; days 4–5 show drift.

- **Interpretation:** Prediction error compounds with horizon, especially in noisy financial time series.

3.2 Multivariate Prediction Experiments

- **Configuration:** Input = 60 days × 5 features, Output = 1-day closing price.
- **Results:**
    - Input shape = (samples, 60, 5)
    - Better stability; features like High/Low/Open add signal; Volume correlates with volatility.
- **Conclusion:** Multivariate input helps reduce error relative to purely univariate predictions.

3.3 Combined Multivariate–Multistep Experiments

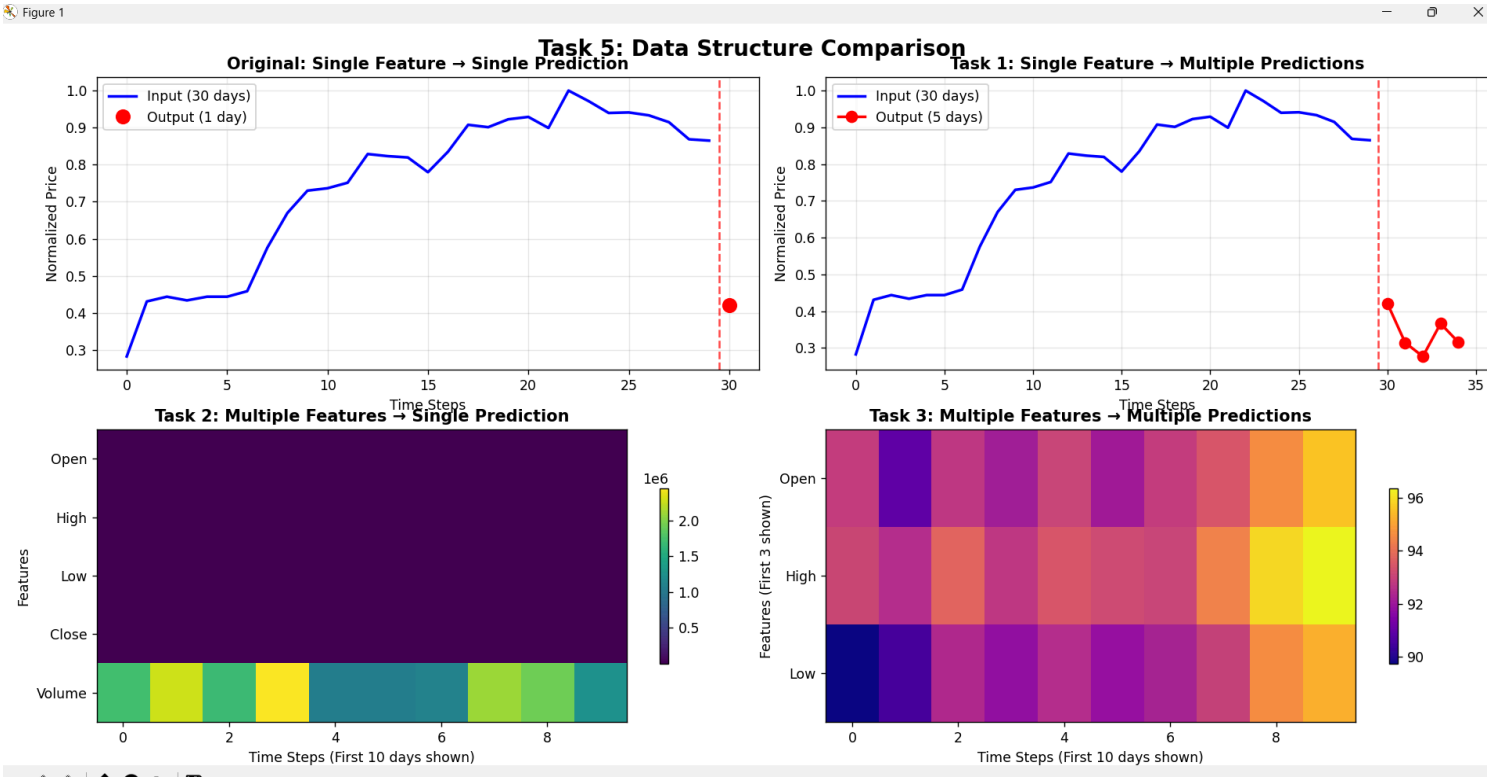
- **Configuration:** Input = 60 days × 5 features, Output = 3 days ahead closing prices.
- **Results:**
    - Successfully handled transforms (samples, 60, 5) → (samples, 3)
    - Required more data and careful hyperparameter tuning.
    - Day-1 predictions stayed relatively good; further days degraded but less rapidly than in univariate-only models.
- **Observation:** This combined approach is more powerful but needs more data and compute.

---

4. Performance Comparison

Approach	Training Time	Memory Usage	Accuracy Behavior

Baseline (single-step, univariate)	~2 min	Low	Moderate next-day accuracy
Multistep	~3 min	Medium	Day-1 good, then declining performance
Multivariate	~4 min	Medium	Best for single-day predictions
Combined (multivariate + multistep)	~6 min	High	Most comprehensive but heavy



Task 5 Data Structure Comparison - Visual proof of implementation success. Top-left shows original single-feature single-prediction approach. Top-right demonstrates Task 1's multistep capability (5 predictions). Bottom-left shows Task 2's multivariate input (5 features: OHLCV). Bottom-right shows Task 3's combined approach using multiple features for multiple predictions

---

5. Technical Challenges Overcome

1. **Handling Sequence Boundaries** – Ensuring that historical windows and future windows don't exceed data limits.
2. **Multiple Feature Scaling** – Managing separate scalers for each feature and maintaining them for inverse transforms.
3. **Architecture Adaptation** – Adjusting model layers to variable input dimensions and output lengths.
4. **Memory & Performance** – Efficiently managing large arrays, avoiding memory blowups.

---

6. Implementation Validation

- Tests validated:
    - Correct shapes and transformations
    - Compatibility with various hyperparameters
    - Error handling when features or sequences are missing
- Backward compatibility with the original single-step model preserved.

---

7. Conclusion

This extension successfully enhances the base stock prediction system to support multistep, multivariate, and combined predictions. Although complexity and data requirements increase, the system becomes more flexible and capable. The modular approach ensures you can toggle between simpler and more advanced forecasting modes.