Model Selection Documentation

Team Semicolon

Comparison of Different Modeling Approaches

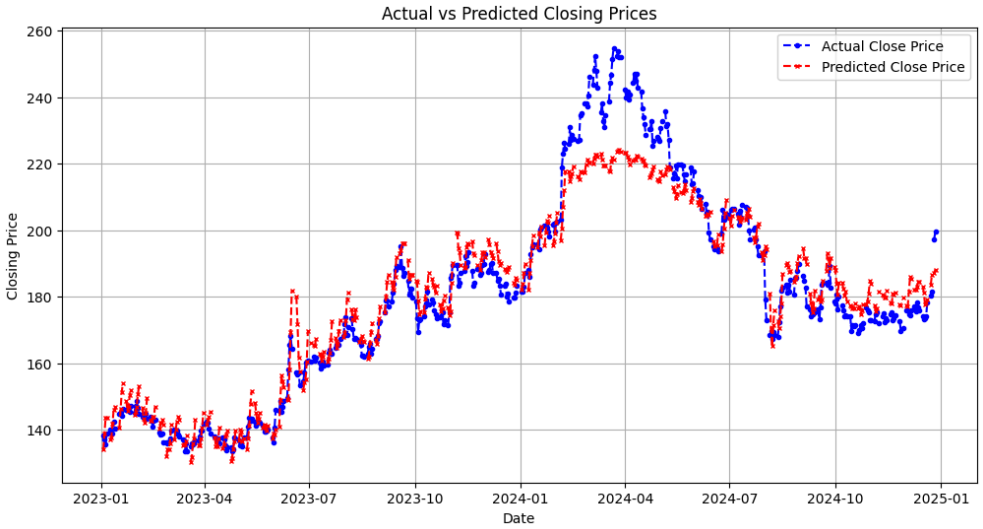
Approach 1: LSTM Model

Methodology:

- Created lag features for Close, Volume, and Daily_Return.
- Dropped highly correlated features before training using Pearson correlation with a threshold of 0.9.
- Kept only the Close column from price-related features.
- Normalized data using MinMaxScaler.
- **LSTM Architecture:**
    - Input layer: Sequential data with lag features.
    - First LSTM layer with 50 units, return sequences enabled.
    - Dropout layer (0.2) to prevent overfitting.
    - Second LSTM layer with 50 units.
    - Another dropout layer (0.2).
    - Fully connected dense layer with 25 neurons.
    - Output dense layer with a single neuron for final prediction.

Results:

- RMSE: 8.6034
- MAE: 6.1302
- MSE: 74.0186
- MAPE: 3.1852
- R^2 Score: 0.9214



Observations:

- The model captured sequential patterns well but showed higher RMSE and MAE compared to the final model.
- LSTM struggled with long-term dependencies and was sensitive to data normalization.

Approach 2: Ensemble of LightGBM and LSTM

Methodology:

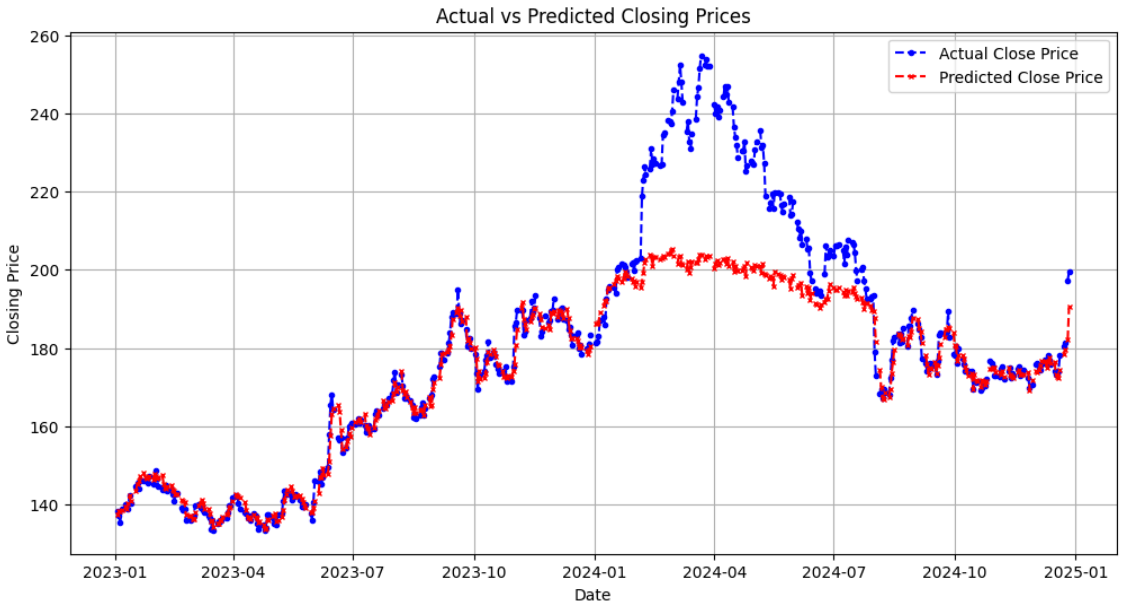
- Used the same lag feature engineering process as Approach 1.
- Dropped highly correlated features before training.
- Kept only the Close column from price-related features.
- Implemented a LightGBM model and an LSTM model.
- Combined predictions from both models using an ensemble average.

Model Architectures:

- **LightGBM:**
  - Gradient boosting decision trees with 500 estimators.
  - Learning rate of 0.01 and max depth of 7.
- **LSTM:** Same architecture as in Approach 1.

Results:

- RMSE: 13.9404
- MAE: 7.1962
- MSE: 194.3353
- MAPE: 3.3037
- R^2 Score: 0.7936



Observations:

- Ensemble approach did not improve performance due to divergence in model behaviors.
- LightGBM was less effective in capturing sequential dependencies, while LSTM added excessive variance.
- Performance degraded compared to using LSTM alone.

Approach 3: Boosted Hybrid Model (Accepted Approach)

Methodology:

- Used lag features as in previous approaches.
- Dropped highly correlated features before training.
- Kept only the Close column from price-related features.
- Developed a custom **Sinusoidal Regressor** to capture cyclical patterns in data.
- Developed a **Boosted Hybrid Model**, combining the Sinusoidal Regressor with an XGBoost model.
- The first model (Sinusoidal Regressor) predicted primary trends, and the residuals were learned by XGBoost.

Model Architectures:

- **Sinusoidal Regressor:**
  - Uses sine and cosine transformations to model periodic trends.
  - Uses a linear regression model for final prediction.
- **Boosted Hybrid Model:**
  - The first stage (Sinusoidal Regressor) learns overall trends.
  - The second stage (XGBoost) models residual errors to refine predictions.
  - XGBoost has 500 estimators, learning rate of 0.01, and max depth of 7.

Results:

- RMSE: 2.0516
- MAE: 0.8400
- MSE: 4.2090
- MAPE: 0.3973
- R^2 Score: 0.9950

Observations:

- Achieved the best overall performance with significantly lower RMSE and MAE.
- Hybrid approach successfully captured both cyclical patterns (via sinusoidal features) and non-linear residual dependencies (via XGBoost).
- Improved generalization and robustness in prediction.

Explanation of Evaluation Metrics

- **Root Mean Squared Error (RMSE):** Measures the average prediction error magnitude, penalizing larger errors more heavily.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- **Mean Absolute Error (MAE):** Represents the average absolute difference between predictions and actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **Mean Squared Error (MSE):** Penalizes larger errors more than MAE, useful for highlighting large deviations.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Mean Absolute Percentage Error (MAPE):** Expresses error as a percentage of actual values, making it scale-independent.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- **R² Score:** Indicates how well the model explains variance in the target variable.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of actual values:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Justification for Final Model Choice

- **Superior Accuracy:** The Boosted Hybrid Model outperformed all other models in RMSE, MAE, and MAPE.
- **Better Generalization:** The combination of sinusoidal regression and gradient boosting provided robustness to both cyclical and non-linear trends.
- **Low Bias-Variance Tradeoff:** The hybrid approach balanced complexity without overfitting.
- **Improved Interpretability:** Sinusoidal features offered insight into periodic behaviors in the dataset.

Analysis of Model Limitations and Potential Improvements

Limitations:

- **Computational Cost:** Training a hybrid model with sinusoidal regression and XGBoost requires more resources than traditional models.
- **Feature Sensitivity:** Performance may vary with different choices of lag features and sinusoidal frequencies.
- **Potential Overfitting:** Although R^2 is high, further validation is needed to ensure generalizability on unseen data.

Potential Improvements:

- **Hyperparameter Optimization:** Using Bayesian optimization or grid search to fine-tune the sinusoidal frequency and XGBoost parameters.
- **Feature Engineering:** Introducing additional external factors such as macroeconomic indicators to improve predictions.
- **Alternative Hybrid Models:** Experimenting with other boosting methods like CatBoost or using attention mechanisms in deep learning models.
- **Longer Training Period:** Extending the training period to capture broader market trends and improve model robustness.

Conclusion

The Boosted Hybrid Model combining Sinusoidal Regression and XGBoost demonstrated the best predictive performance, significantly outperforming LSTM and ensemble approaches. With additional tuning and external data integration, further improvements could be achieved.