“HYBRID” MODEL FOR STOCK PRICING PREDICTION

CHI NGUYEN

PRACTICUM I

CONTENT

- PROJECT INSPIRATION
- QUESTIONS TO ANSWER
- DATA SOURCE & MACHINE LEARNING METHODS
- PROJECT STEPS
- LSTM MODEL PERFORMANCE
- HYBRID MODEL PERFORMANCE
- CONCLUSION

PROJECT INSPIRATION

- Stock price prediction has been one of my very first project idea when I chose Data Science for my master's degree.
- "Hybrid" model is an idea from an article on Neptune.ai about adding MA predictions as input vectors to the LSTM model to improve the prediction performance.

Ref: https://neptune.ai/blog/predicting-stock-prices-using-machine-learning

QUESTIONS TO ANSWER

- How well machine learning can help to predict stock pricing?
- How feature selection may affect model performance?
- How “hybrid” model can be better than normal LSTM model?

DATA SOURCE & MACHINE LEARNING METHODS

- All data including: Stock data, Economics data, Techinical indicators are all extracted via API from Alphavantage.co
- This page has document to guide us to pull data using their API.
- The data is collected in “full” method which means all the data in 20 years.
- LSTM is known as one of the most popular and effective models to predict stock price. In this project, I will explore it with an idea of “hybrid model” inspired by the article I mentioned before.

OVERALL PROJECT STEPS

1. DATA COLLECTING & CLEANING

- Stock data: Nike company
- Economics data: Inflation, Unemployee rate, Retail sales value (millions of dollar), CPI (Consumer Price Index), Federal funds rate, treasury yield.
- Technical indicators: MACD, RSI, BBANDS, STOCH

2. FEATURE SELECTION

Use 2 methods: Forward Feature Selection(FFS), Recursive Feature Elimination (RFE).

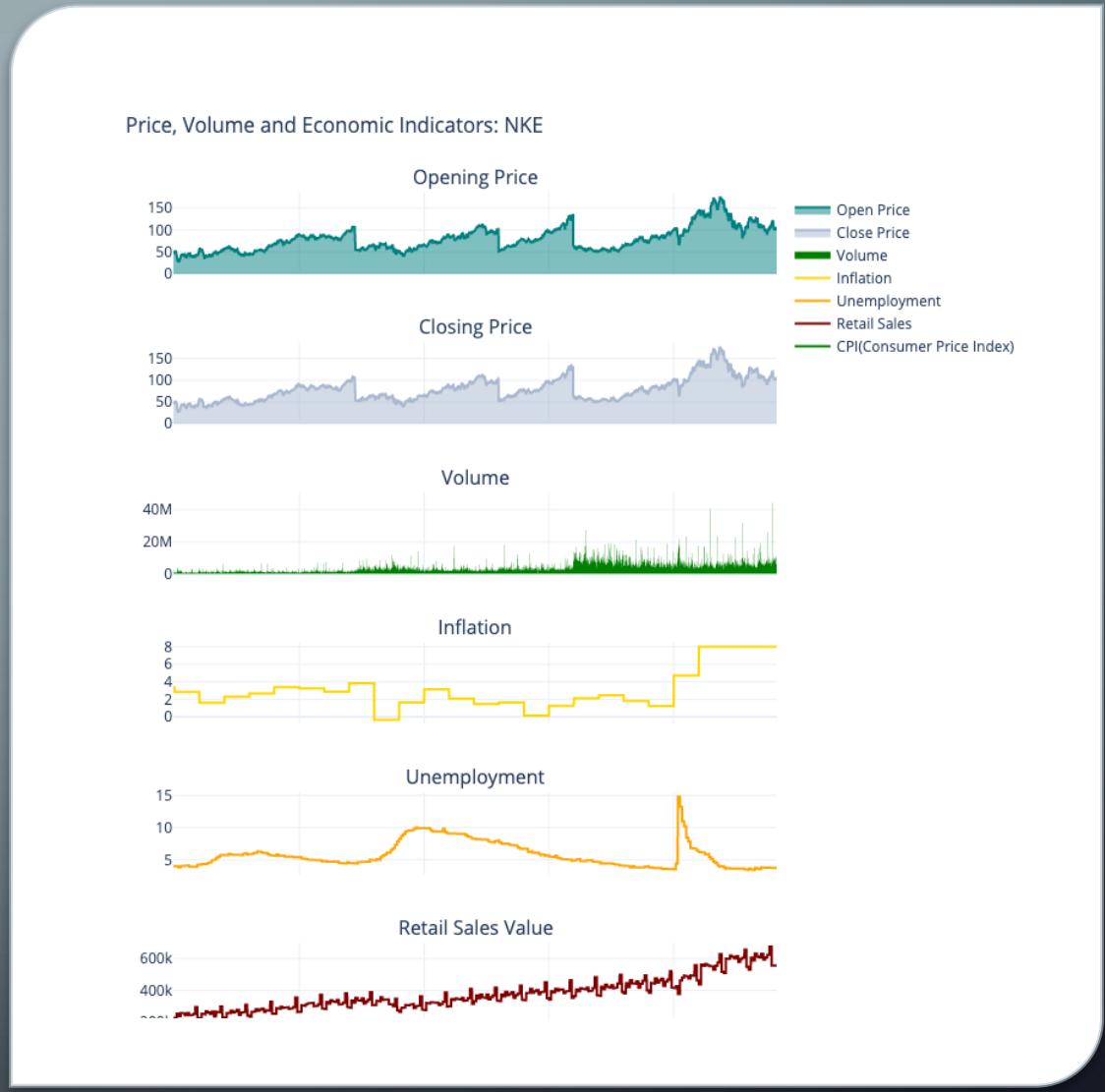
3.LSTM MODELS & HYBRID MODEL

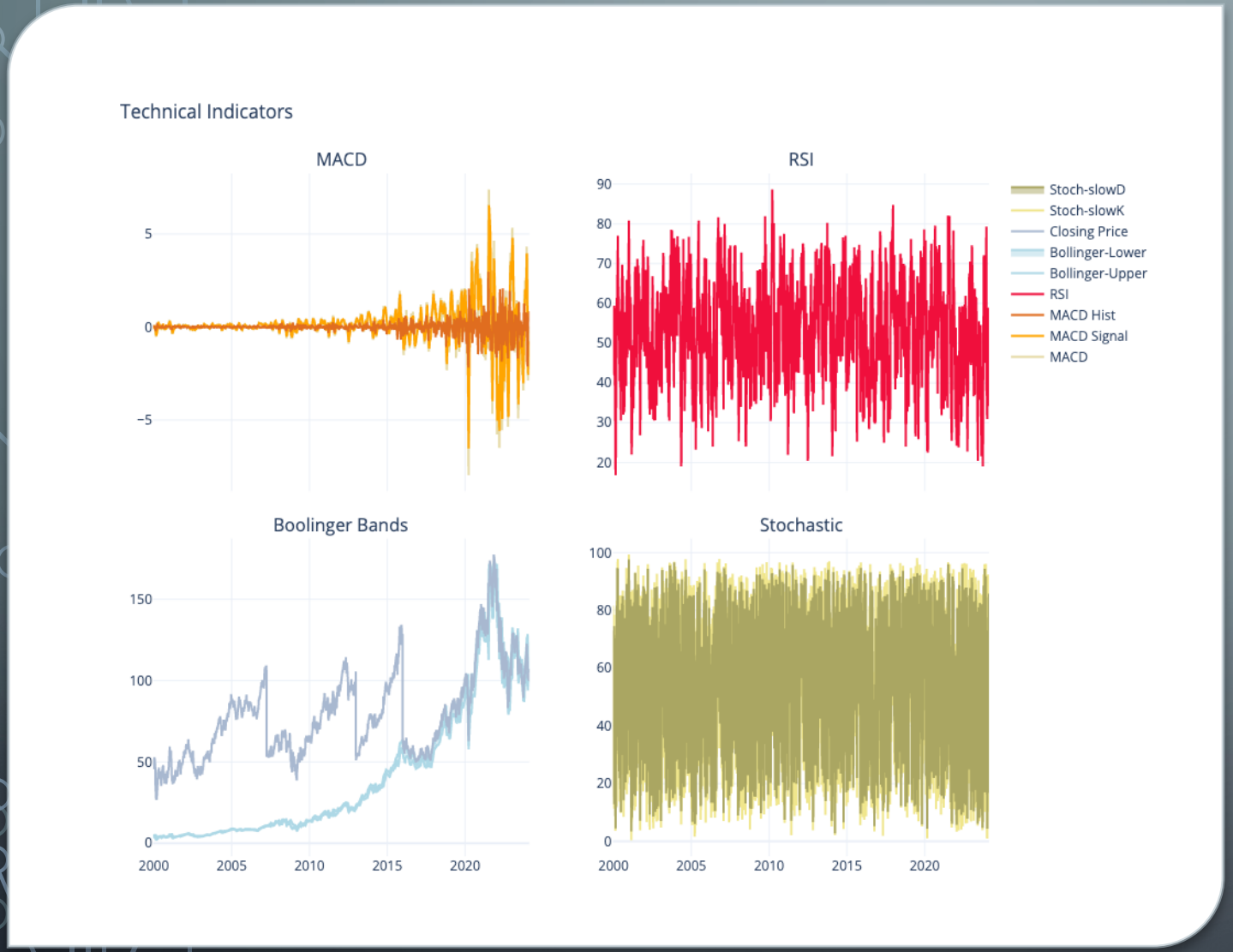
LSTM models:

- 1 use data having features selected by Forward Feature Selection method
- 1 by RFE method

The model having better performance will be the chosen one to develop the “hybrid” model.

HAVE A LOOK AT THE DATA:
PRICE, VOLUME OF NIKE AND ECONOMIC INDICATORS OVER 20 YEARS



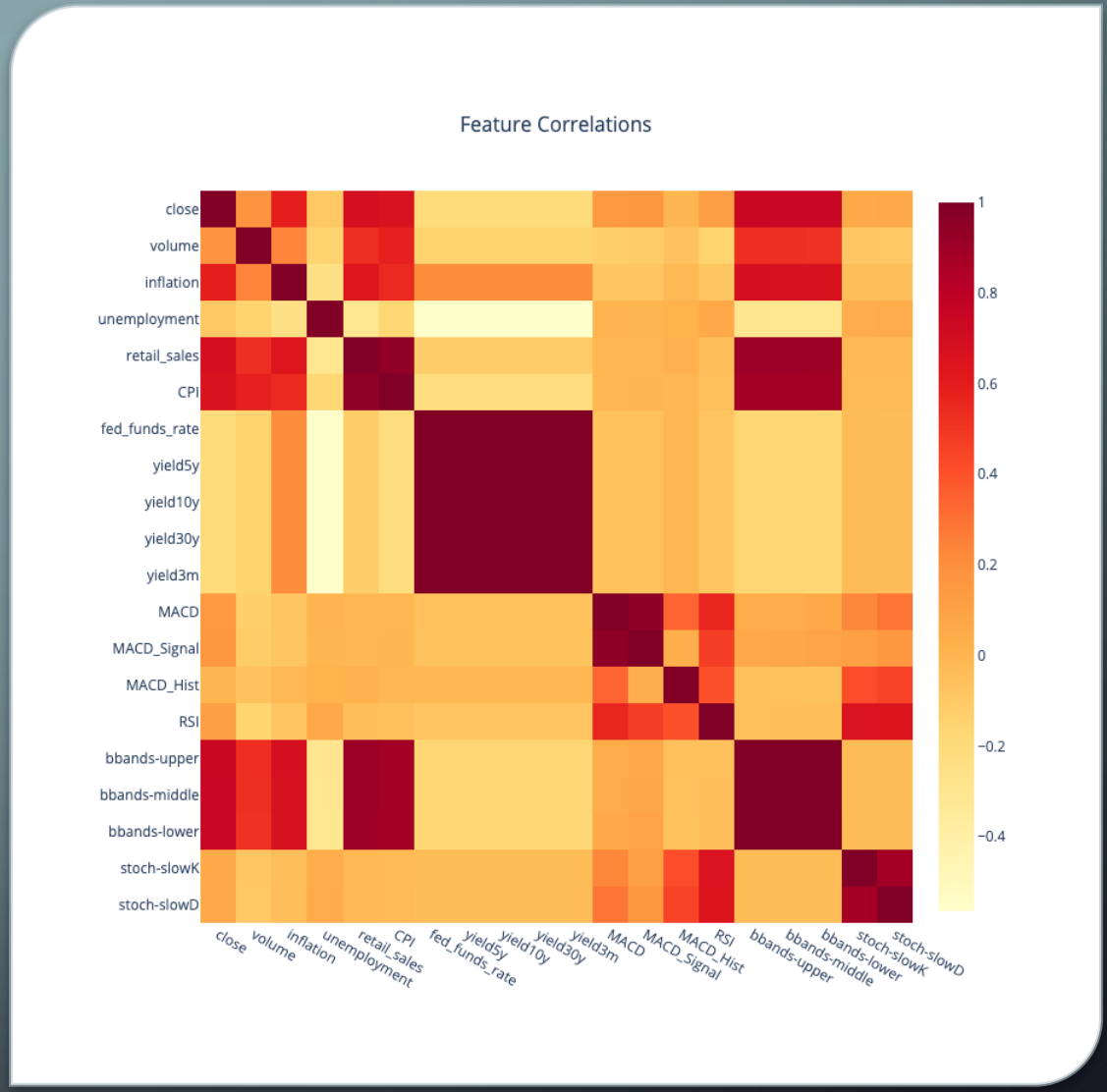


HAVE A LOOK AT THE TECHNICAL INDICATORS

CORRELATION HEATMAP

Economic indicators such as Inflation, Retail_sales, CPI and Technical indicators such as MACD, RSI, BBANDS seem to have high correlation to the ‘close’ price.

Let’s check and select features with FFS and RFE.



FEATURE SELECTION RESULT BY OUR 2 METHODS:

- **FORWARD FEATURE SELECTION (FFS):** 'volume', 'inflation', 'unemployment', 'retail_sales', 'CPI', 'MACD', 'MACD_Signal', 'RSI', 'stoch-slowK', 'stoch-slowD'
- **RECURSIVE FEATURE ELIMINATION (RFE):** 'inflation', 'unemployment', 'CPI', 'fed_funds_rate', 'yield10y', 'yield3m', 'MACD', 'bbands-upper', 'bbands-middle', 'bbands-lower'

DATA PREPROCESSING

Clean the data

- Change the column names
- Drop the missing data

Transform to stationary

- Non-stationary data can lead to unreliable model outputs and inaccurate predictions, just because the models aren't expecting it.
- Augmented Dickey Fuller Test (ADF Test) to test whether a given Time series is stationary or not

Prepare the data

- Reverse the date-time index so that the data will be in the ascending order and easier for the model to understand the data
- Convert the date-time index to integer
- Convert to numpy arrays as LSTM model requirement

Scale the input, output

- As we can see in the charts shown previously, stock market with a wide range of values in data. This may result in large error gradient values in the machine learning process.
- That's why we need to scale the input, output here.

Split train,test data

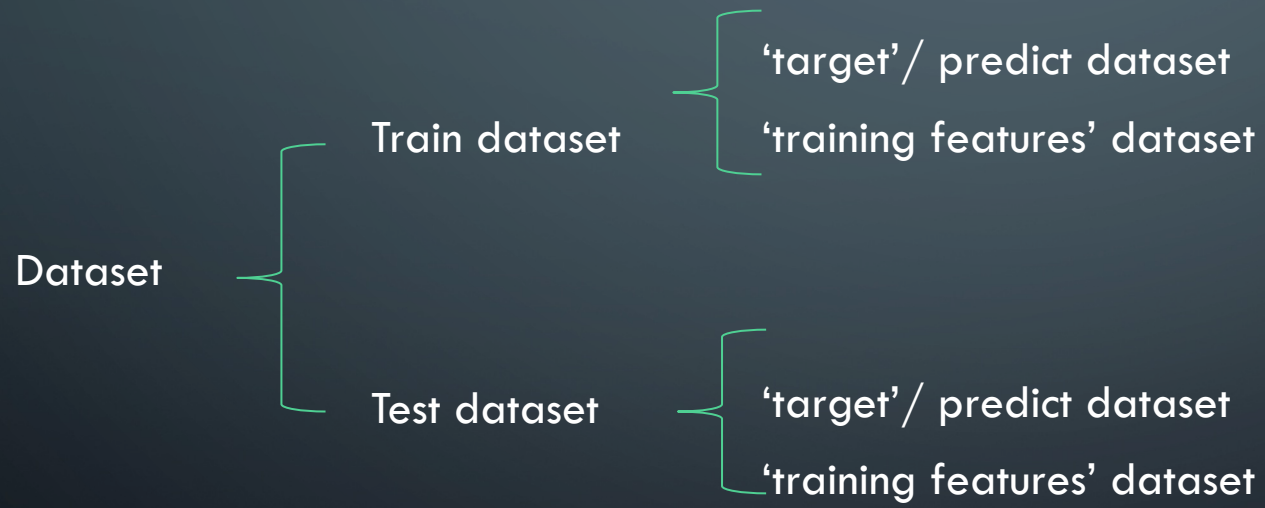
- Split percentage: 80%

Partition the train,test data

- Partition the data to use one subset of data to the model, and another subset to test how well the model works on new and unseen data.

THE DATA FOR LSTM MODEL

- First, I split the data into Train, Test dataset
- Then, I partitioned each Train, Test dataset into ‘target’ and ‘training features’



LSTM MODEL – FFS METHOD

Is the LSTM using the dataset with features selected by FFS method

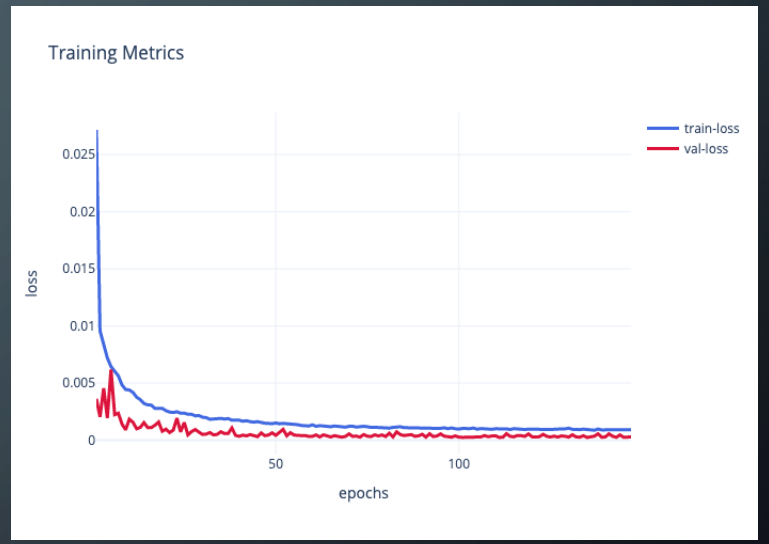
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 25, 25)	4300
lstm_1 (LSTM)	(None, 25, 256)	288768
dropout (Dropout)	(None, 25, 256)	0
lstm_2 (LSTM)	(None, 128)	197120
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 32)	4128
dropout_2 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dropout_3 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 3)	51

...

Total params: 494895 (1.89 MB)

Trainable params: 494895 (1.89 MB)

Non-trainable params: 0 (0.00 Byte)



Model Error:

Mean Absolute Error (MAE): 0.15

Root Mean Squared Error (MSE): 0.18

Mean Absolute Percentage Error (MAPE): 3.15%

Median Absolute Percentage Error (MDAPE): 3.0%

LSTM MODEL – RFE METHOD

Is the LSTM using the dataset with features selected by RFE method

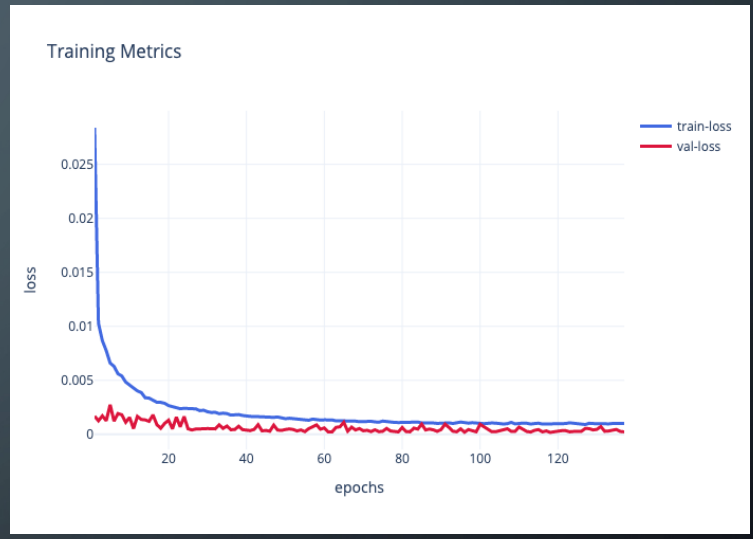
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 25, 25)	4400
lstm_1 (LSTM)	(None, 25, 256)	288768
dropout (Dropout)	(None, 25, 256)	0
lstm_2 (LSTM)	(None, 128)	197120
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 32)	4128
dropout_2 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dropout_3 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 3)	51

...

Total params: 494995 (1.89 MB)

Trainable params: 494995 (1.89 MB)

Non-trainable params: 0 (0.00 Byte)



Model Error:

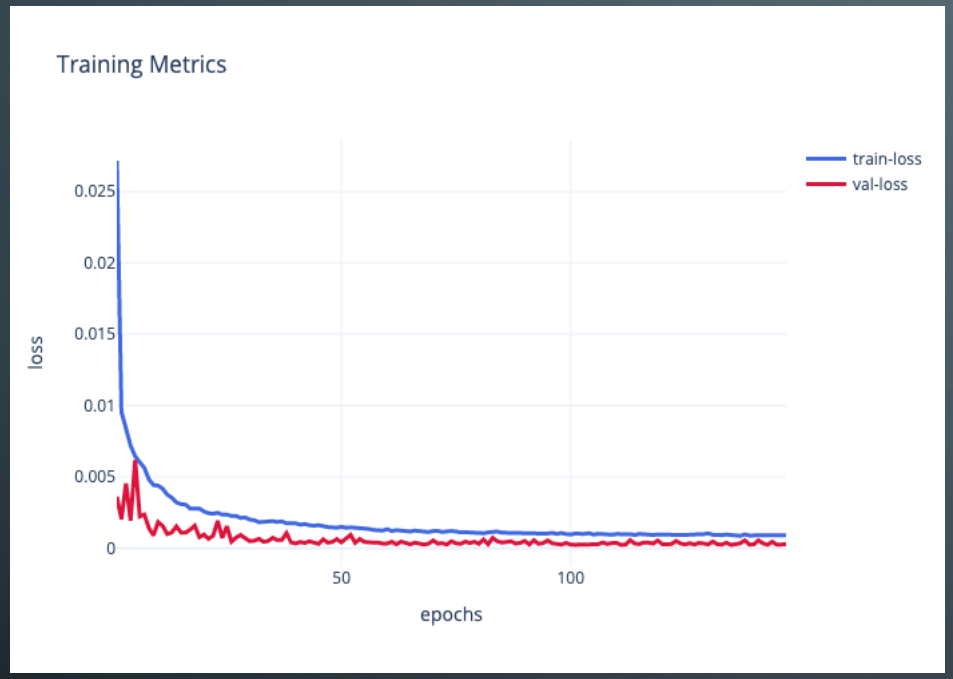
Mean Absolute Error (MAE): 0.08

Root Mean Squared Error (MSE): 0.09

Mean Absolute Percentage Error (MAPE): 1.65%

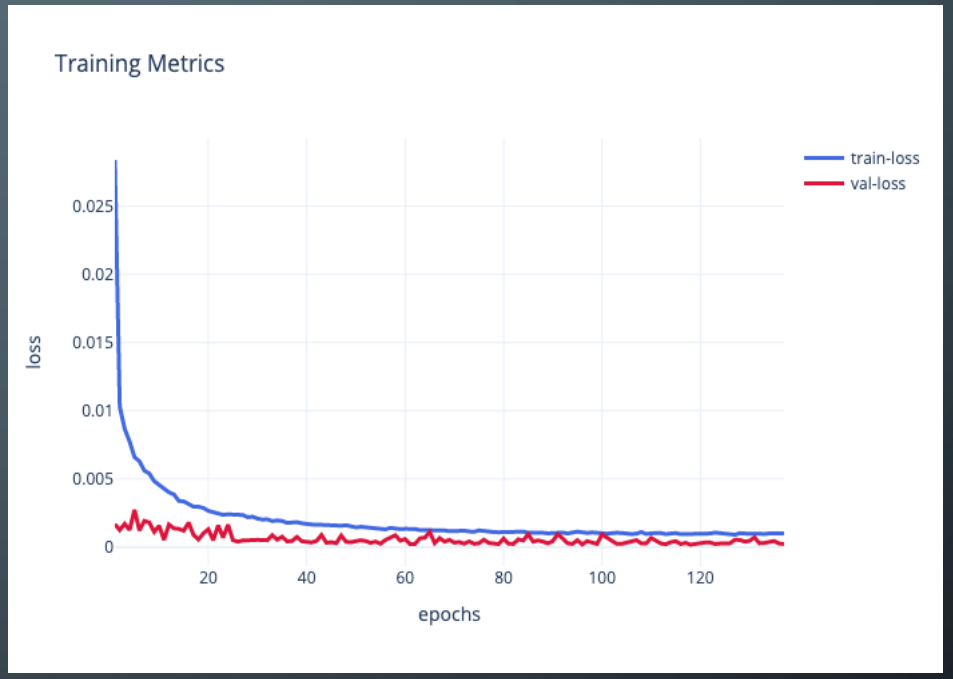
Median Absolute Percentage Error (MDAPE): 1.63%

LSTM MODEL PERFORMANCE



FFS

Mean Absolute Error (MAE): 0.15
Root Mean Squared Error (MSE): 0.18
Mean Absolute Percentage Error (MAPE): 3.15%
Median Absolute Percentage Error (MDAPE): 3.0%



RFE

Mean Absolute Error (MAE): 0.08
Root Mean Squared Error (MSE): 0.09
Mean Absolute Percentage Error (MAPE): 1.65%
Median Absolute Percentage Error (MDAPE): 1.63%

Better score -> move forward with this method

HYBRID MODEL WITH MA PREDICTIONS AS THE INPUT VECTORS

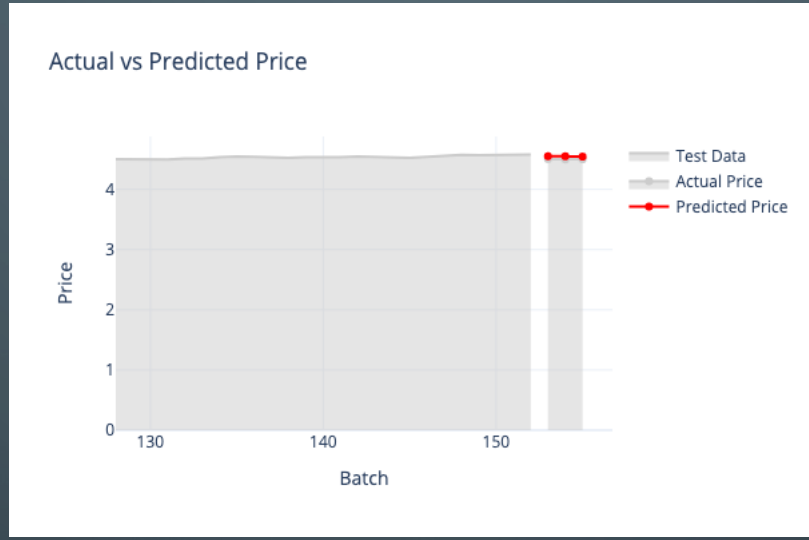
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 25, 25)	4500
lstm_1 (LSTM)	(None, 25, 256)	288768
dropout (Dropout)	(None, 25, 256)	0
lstm_2 (LSTM)	(None, 128)	197120
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 32)	4128
dropout_2 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dropout_3 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 3)	51

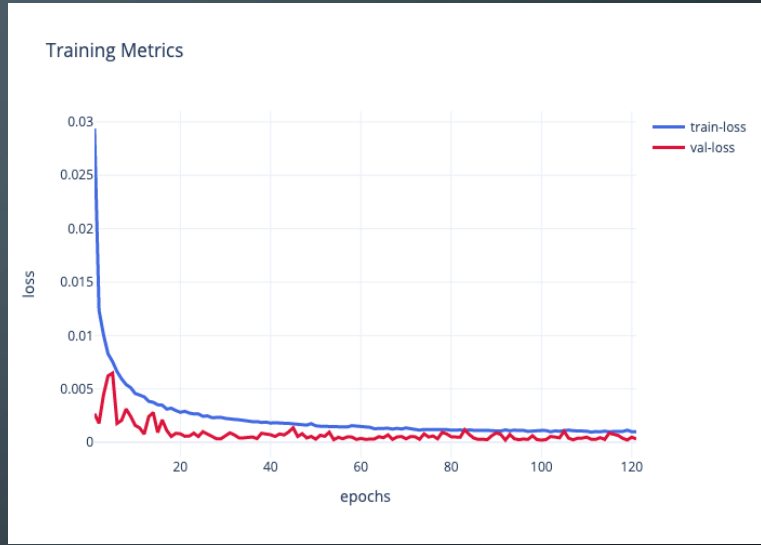
...

Total params: 495095 (1.89 MB)

Trainable params: 495095 (1.89 MB)

Non-trainable params: 0 (0.00 Byte)





Mean Absolute Error (MAE): 0.06

Root Mean Squared Error (MSE): 0.07

Mean Absolute Percentage Error (MAPE): 1.17%

Median Absolute Percentage Error (MDAPE): 1.01%

CONCLUSION

How well machine learning can help to predict stock pricing?
How feature selection may affect model performance?
How “hybrid” model can be better than normal LSTM model?

- LSTM is obviously powerful model in predicting price with really good scores.
- Choosing feature selection method is an important step affecting the model performance. Data professionals should perform different methods to be able to see the difference, then pick the more suitable method for the dataset.
- ‘Hybrid’ model delivered better performance which can be potential idea for further improvement. For example, add more input vectors the same as MA predictions to enhance the LSTM model.

For the further steps: Hyperparameter tune step can be taken to see if the performance can be improved.

REFERENCE

- Lagged features: https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/
- Input & Output shape of LSTM: https://www.kaggle.com/code/shivajbd/input-and-output-shape-in-lstm-keras
- Partition data: https://www.linkedin.com/advice/3/what-best-ways-partition-data-training-testing#:~:text=Partitioning%20data%20into%20training%20and,performance%20in%20real%2Dworld%20scenarios
- Stationary data: https://hex.tech/blog/stationarity-in-time-series/#:~:text=Non%2Dstationary%20data%20can%20lead,than%20non%2Dstationary%20time%20series
- ADF test: Ref: https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/
- Partition data: https://www.linkedin.com/advice/3/what-best-ways-partition-data-training-testing#:~:text=Partitioning%20data%20into%20training%20and,performance%20in%20real%2Dworld%20scenario
- Code reference: https://github.com/kconstable/market_predictions/tree/main