Predicting coffee price

Time series analysis

Yair Haendler

Data Science - Capstone project

General Assembly - August 2021

Background and problem statement

Mysterious origins

Yemen?

Ethiopia?

Mysterious origins

Reliable evidence of first coffee roasting and brewing: **Yemen, mid 15th Century**.

16th Cent.: reached the Middle East and **Turkey**.

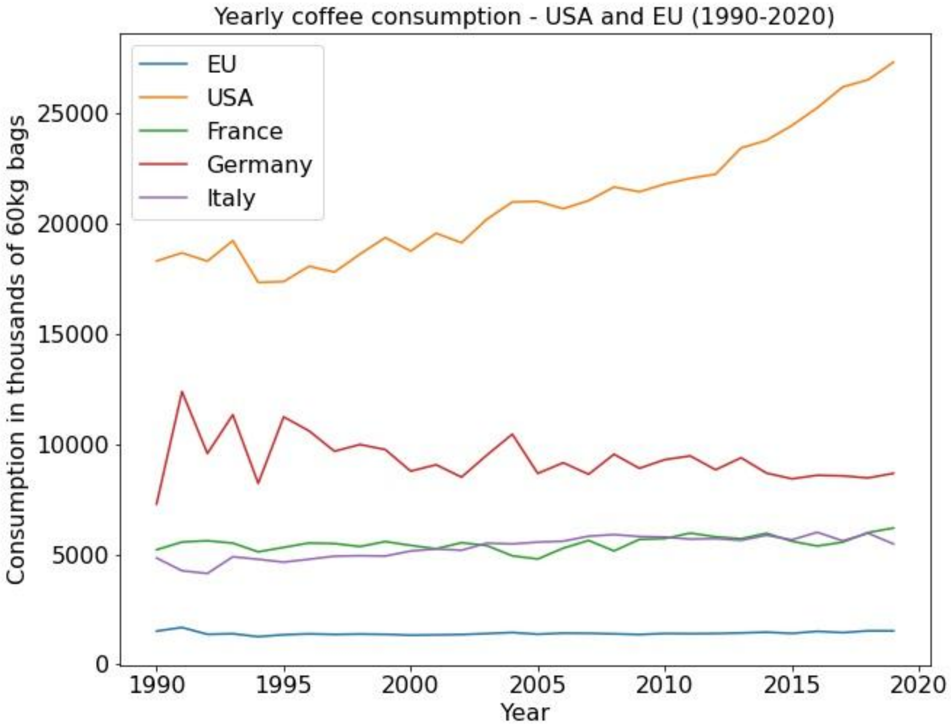
Then: Venice and the rest of **Europe**.

https://en.wikipedia.org/wiki/Coffee#History

Fascinating story

One of the most in-demand commodities world-wide.

Some numbers



What factors affect coffee prices

- Supply and demand (lower production → higher prices).
- Weather conditions.
- Political changes.
- Currency fluctuations.
- Commodity trading rules.
- ...

Problem statement

- Goal: forecasting **coffee price** based on historical data.

- Absence of additional predictors (**Univariate Time Series Forecast**).

Data collection - challenges and resolutions

Data sources

- **International Coffee Organization**:

  - Yearly data - historical (1990-2019); divided by country.

  - Daily data - recent (2020-2021); average of France, Germany and USA.

  - Non-price data (exports, imports, consumption, trade, etc.): yearly data (1990-2019); divided by country.

Data sources

- **USDA - PSD**:

  - Half-yearly / yearly data - historical & recent data (~1960 - 2020).

  - Average international market prices.

Data sources

- **macrotrends.net**:

  - Daily data - historical & recent data (August 1973 - August 2021).

  - Average international market prices.

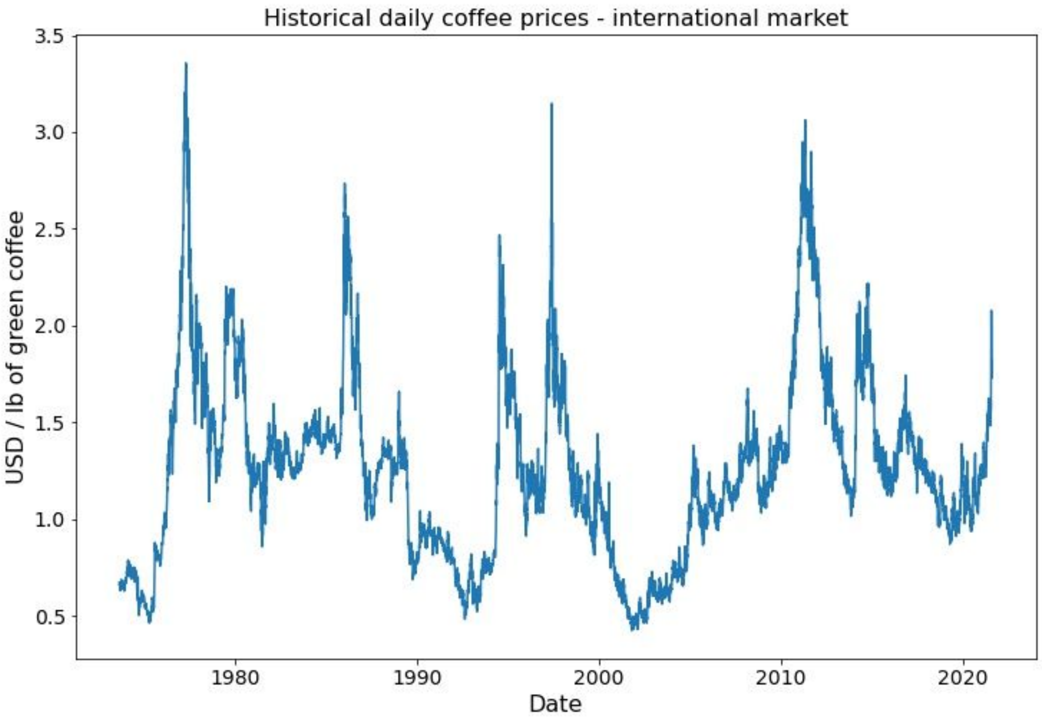
  - **Pro:** more data → better forecast ability.

  - **Con:** no information broken by country; no external predictors.

Data preparation

- Organized the data in one ***Price*** column.
- Set the ***Date*** as the (sorted) index.
- Checked for missing values.
- Some ‘gaps’ - dates without data (our models can handle this).

The data



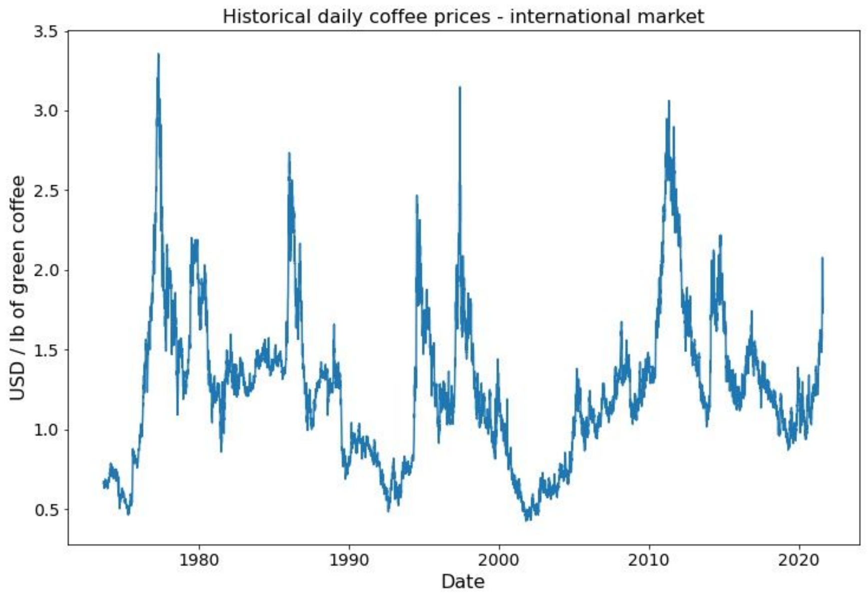
Baseline scores:

MSE: 1.585

MAPE: (Mean Absolute Percentage Error) **115.73%**

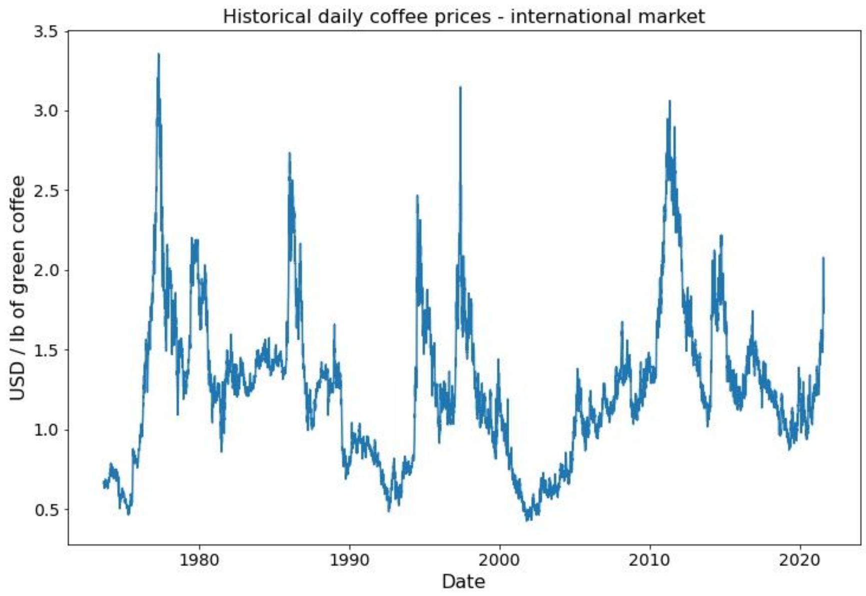
The forecasting challenge

- Data is stationary (ADF test: p = 0.0061).
- No trend?
- No seasonality.



The forecasting challenge

- Coffee doesn’t have general “seasons”; it is consumed according to necessity of the **individual**.
- It is consumed always, all the time (hot AND cold).



Modeling

ARIMA

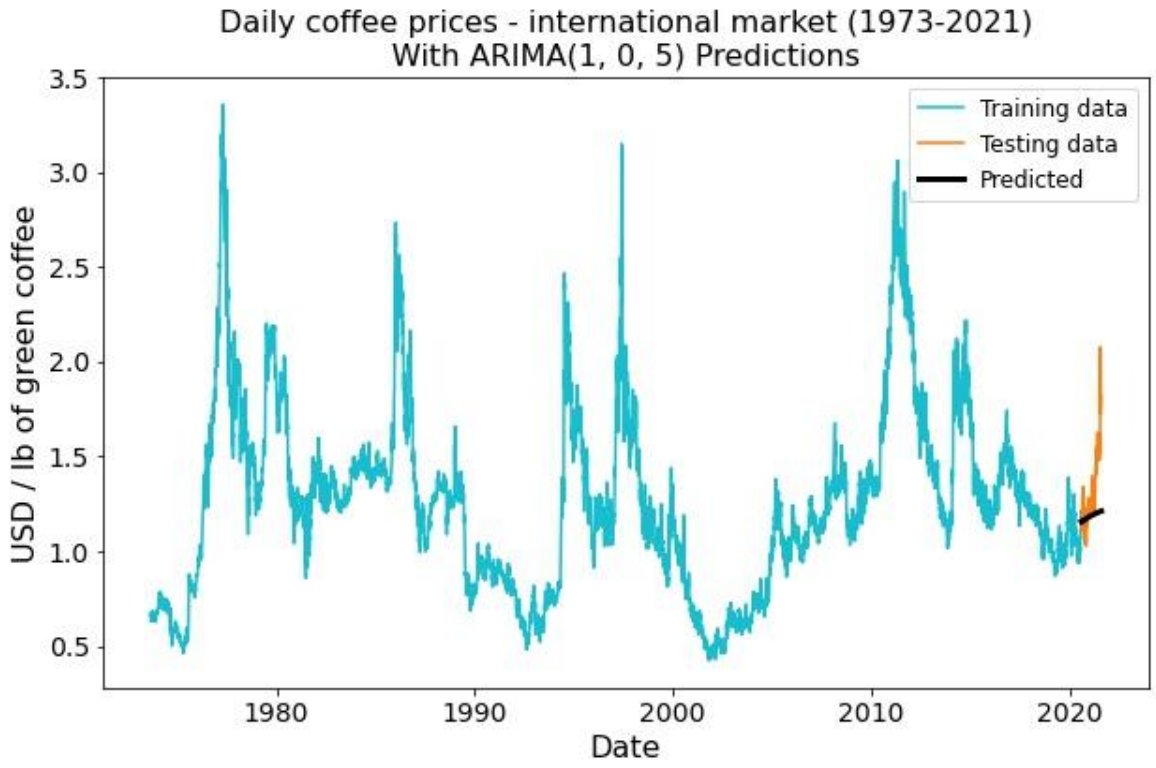
ARIMA

- **ARIMA (1, 0, 5)** - gridsearch to find best hyperparameters

- **Train data** → all the data up to the last year

  Test data → the last year (259 days due to missing data)

ARIMA's results



(Baseline MSE: 1.585)

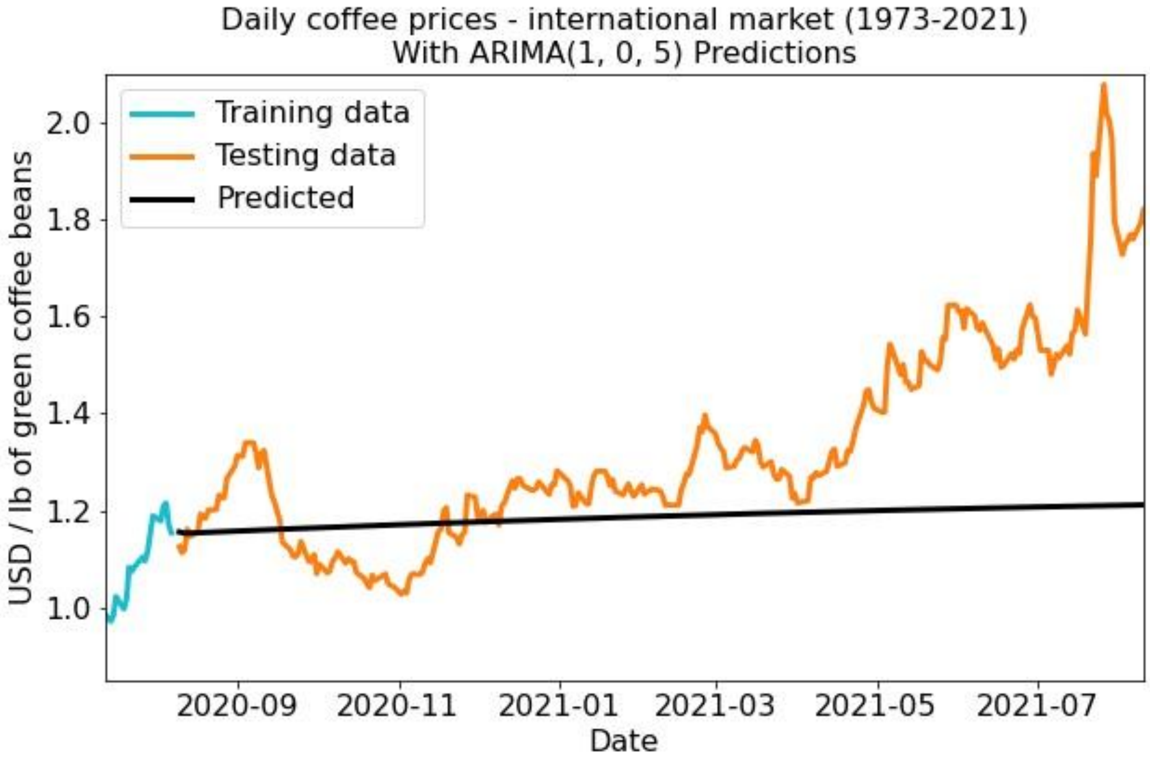
Training MSE: 0.0009

Training MAPE: 47.70%

Testing MSE: 0.056

Testing MAPE: 12.17%

ARIMA's results



(Baseline MSE: 1.585)

Training MSE: 0.0009

Training MAPE: 47.70%

Testing MSE: 0.056

Testing MAPE: 12.17%

Prophet

ARIMA vs. Prophet

- **ARIMA** assumes <u>causal relationship</u> between **past values and past error**, and **future values**.

- **Prophet** assumes <u>no causal relationship</u> between past and future values.

ARIMA vs. Prophet

- Best curve (linear / logistic) fitted to the data.

- Naïve assumptions concerning changes in **trend** and **seasonality** - “what happened in the past will happen again in the future”.

  Useful with stable trend and seasonality.

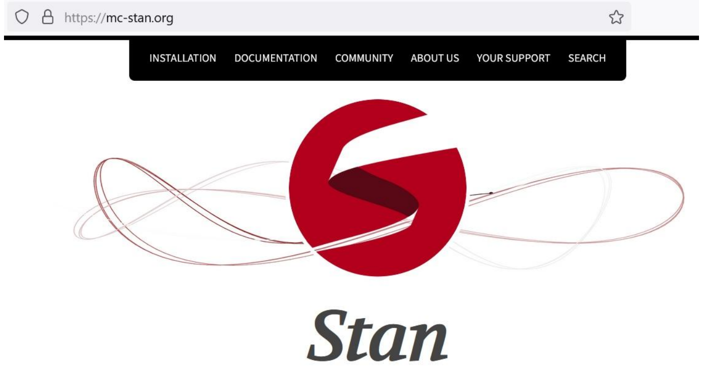
ARIMA vs. Prophet

- <u>The Prophet model:</u>

$$y(t) = g(t) + s(t) + h(t) + e_t$$

Trend **Seasonality** **Holidays**

Prophet



https://facebook.github.io/prophet/

Trend - changepoint detection

- Prophet sets automatically **changepoints** - points of potential trend change.
- **Sparse prior** (with many zeros) for the magnitude of these possible changes.
- Only meaningful rate changes determine future trend shape.

Adjusting the changepoints prior

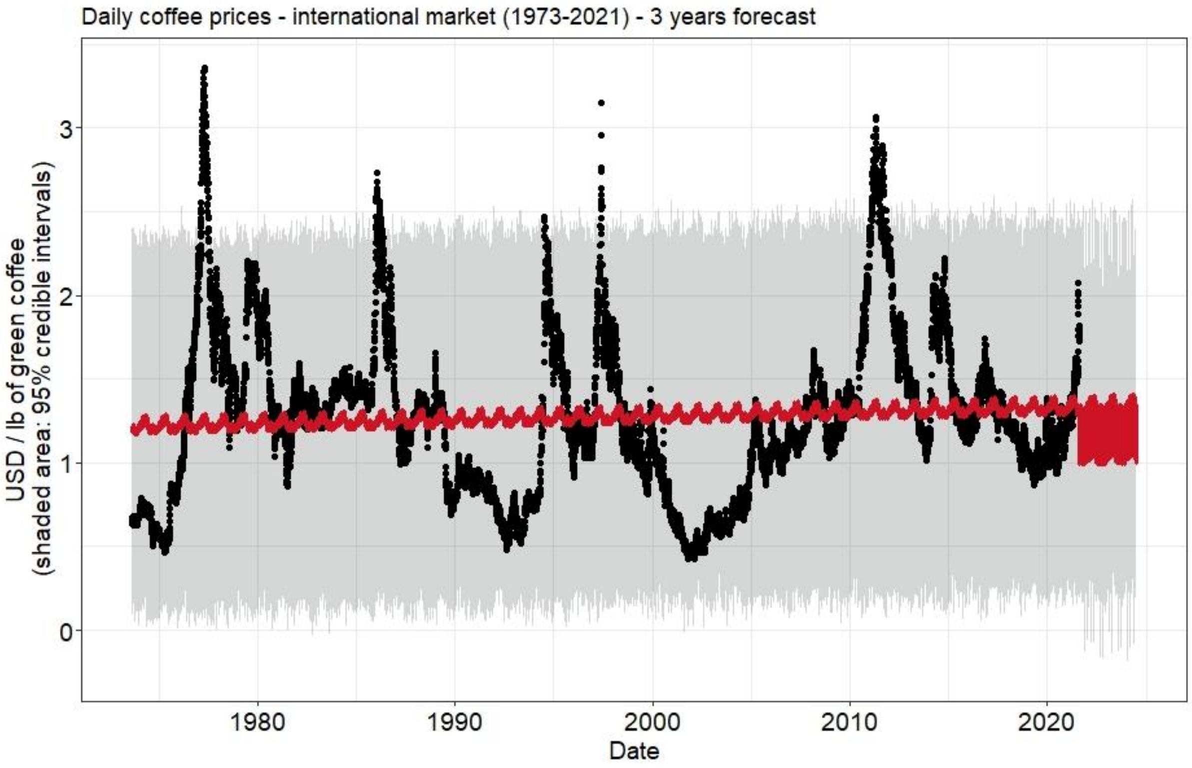
- Sparse prior adjustment deals with trend overfitting or underfitting.

- More changepoints → **more fluctuations** in the predictions.

  Less changepoints → **less fluctuation** in the predictions.

- “What happened in the past will happen again in the future”

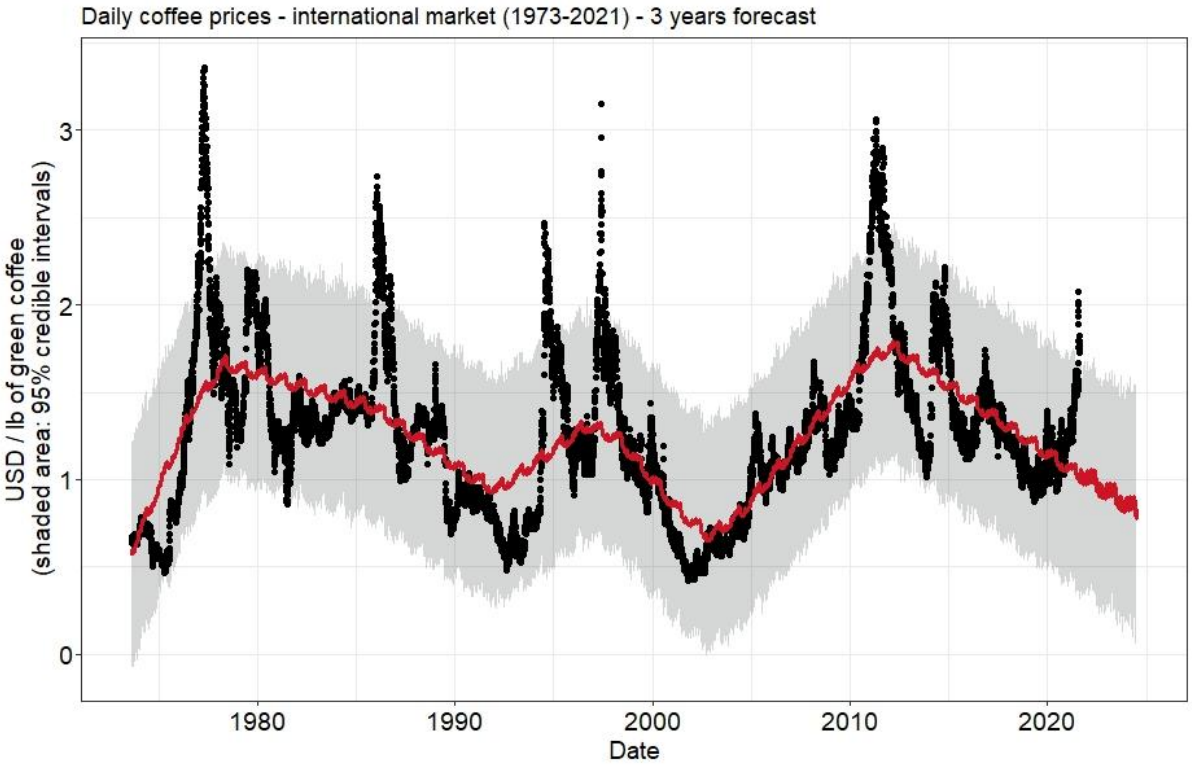
Adjusting the changepoints prior



Changepoint prior scale:

0.0001

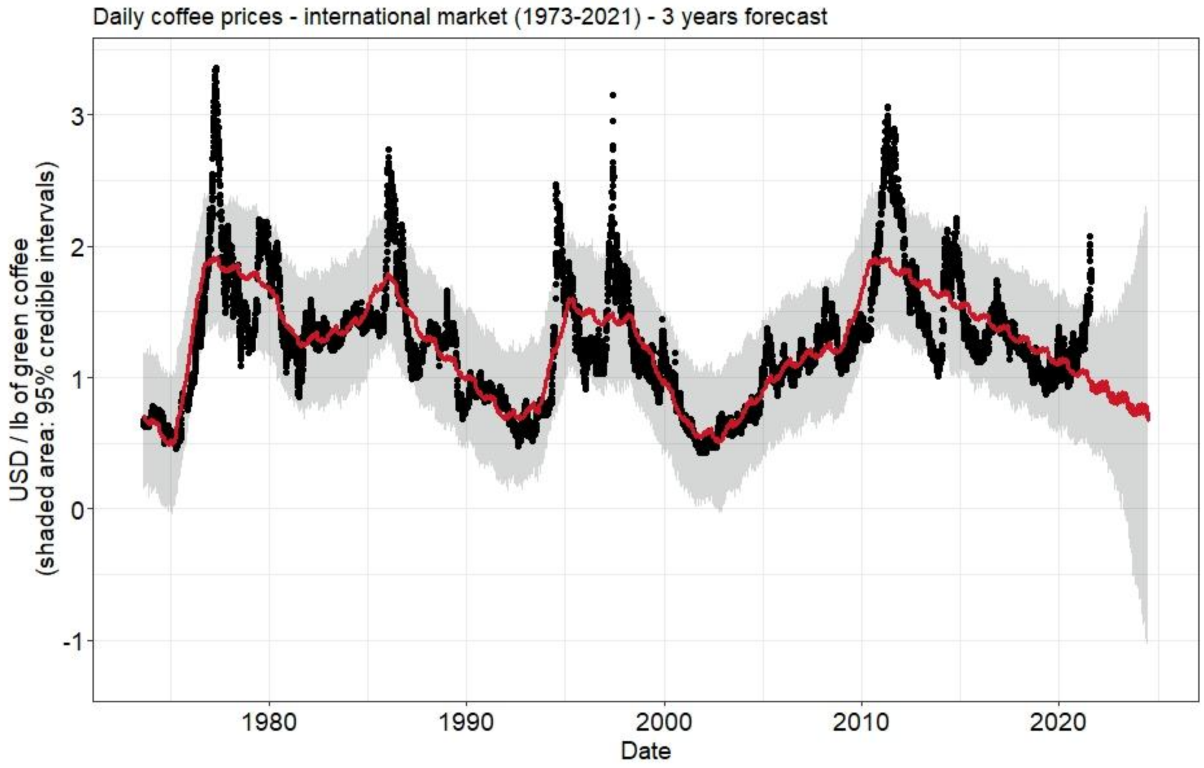
Adjusting the changepoints prior



Changepoint prior scale:

0.01

Adjusting the changepoints prior



Changepoint prior scale:

0.1

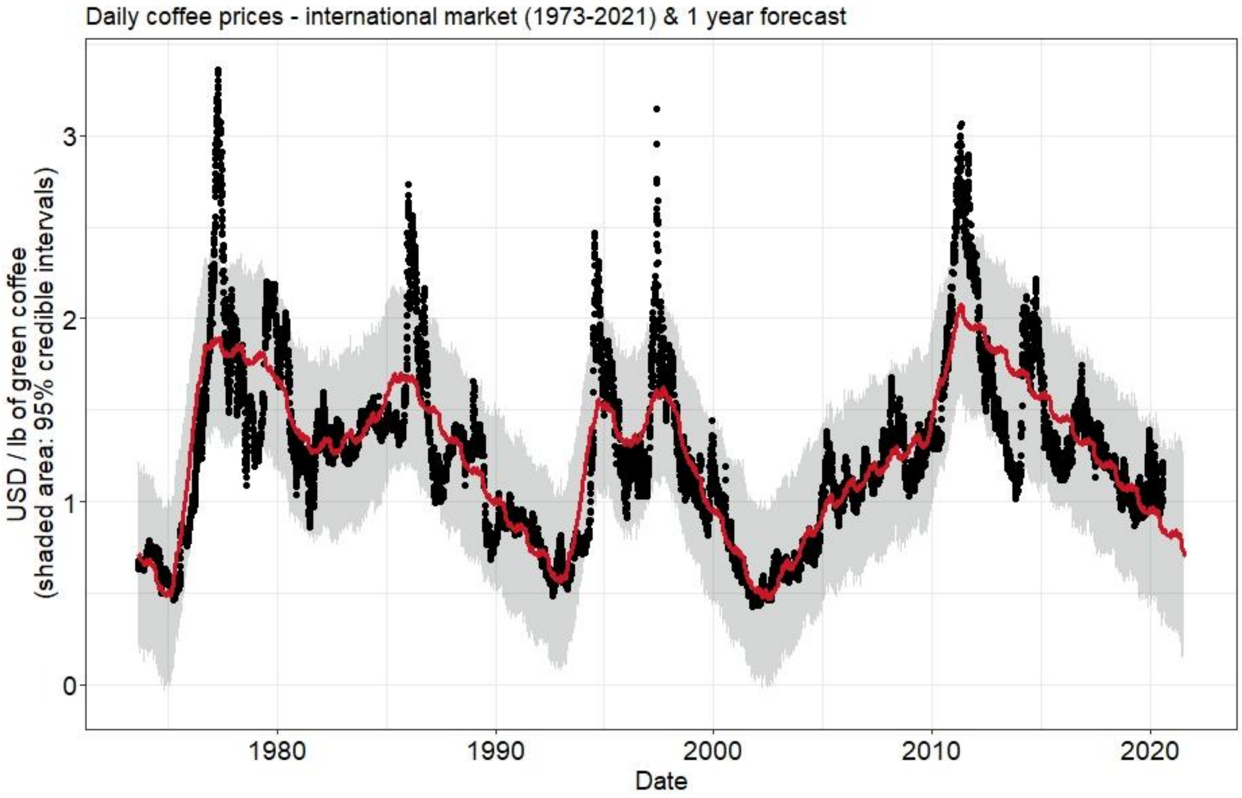
Prophet

- **Changepoint prior scale = 0.1**

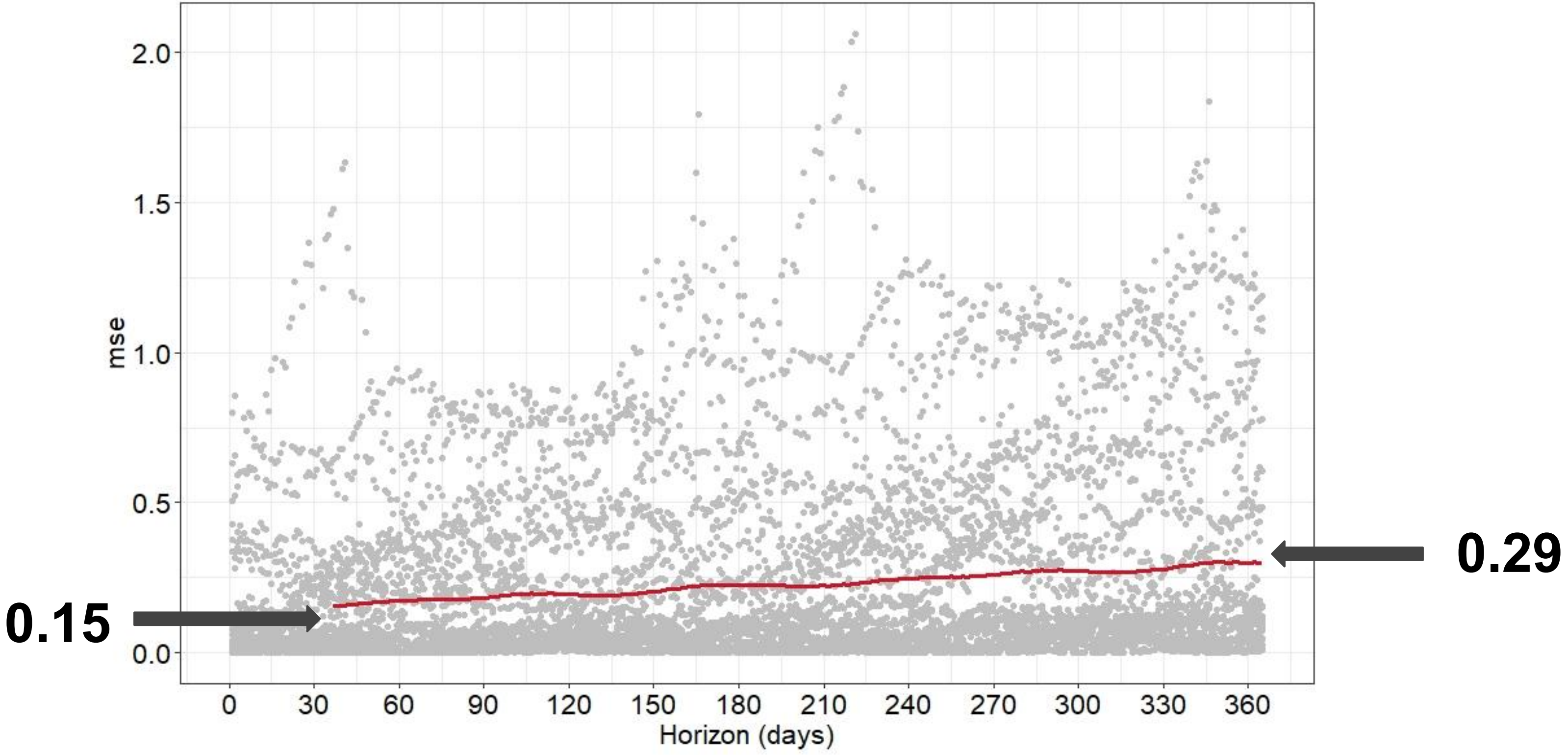
  Seasonality prior scale = 0.01

- Same Train-Test split as with the ARIMA model

Prophet’s results

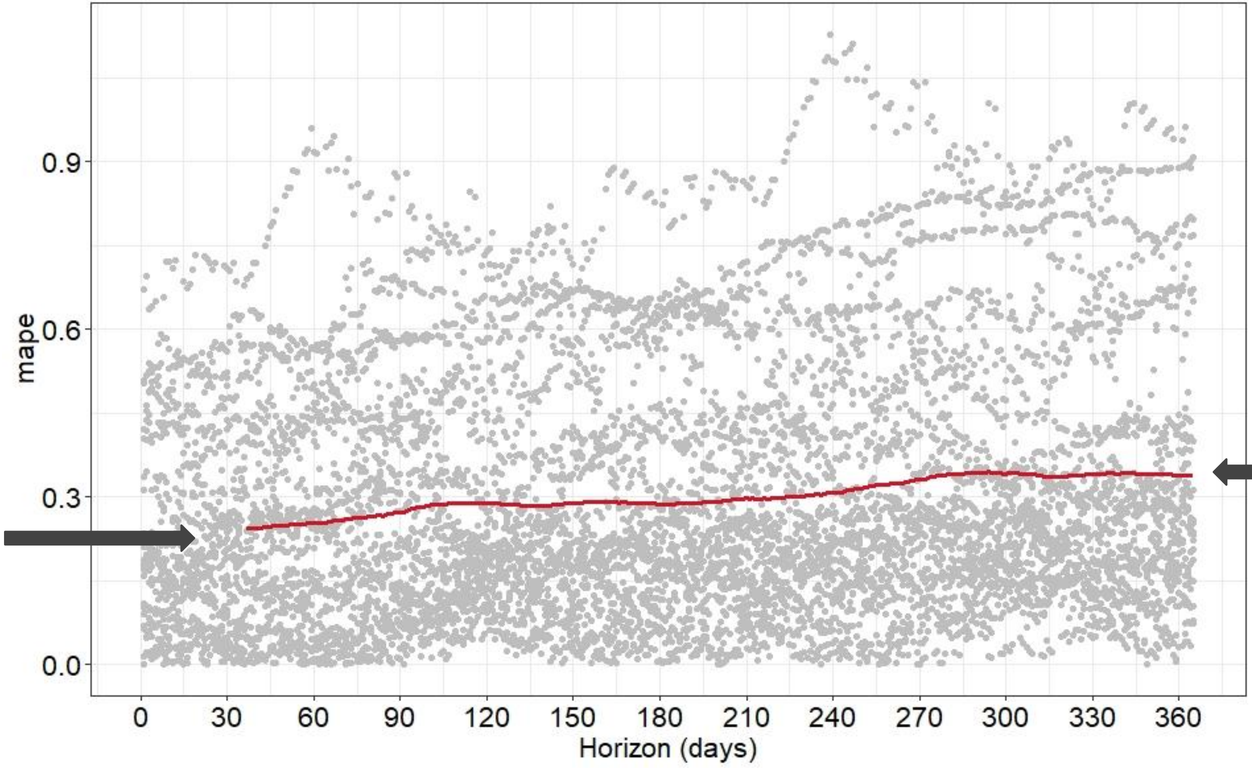


Simulated Historical Forecasts - MSE



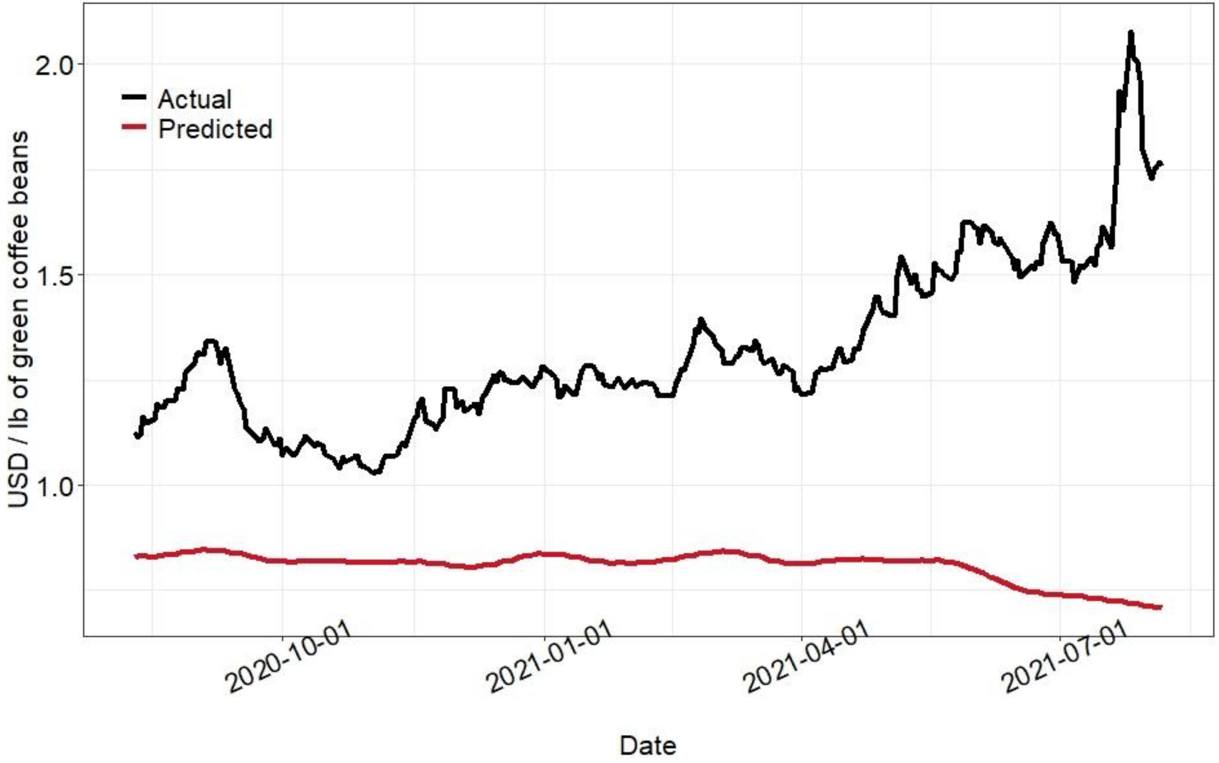
Simulated Historical Forecasts - MAPE

24%



33.8%

Testing data - predictions vs. actual prices



Testing MSE:

0.317

Testing MAPE:

37.36%

Recurrent Neural Networks

Recurrent Neural Networks

- Number of input values = 10

  Each 10 values predict the next value:

  [0,1,2,3,4,5,6,7,8,9] => [10]
  [1,2,3,4,5,6,7,8,9,10] => [11]
  …

- Train / Validation / Test split

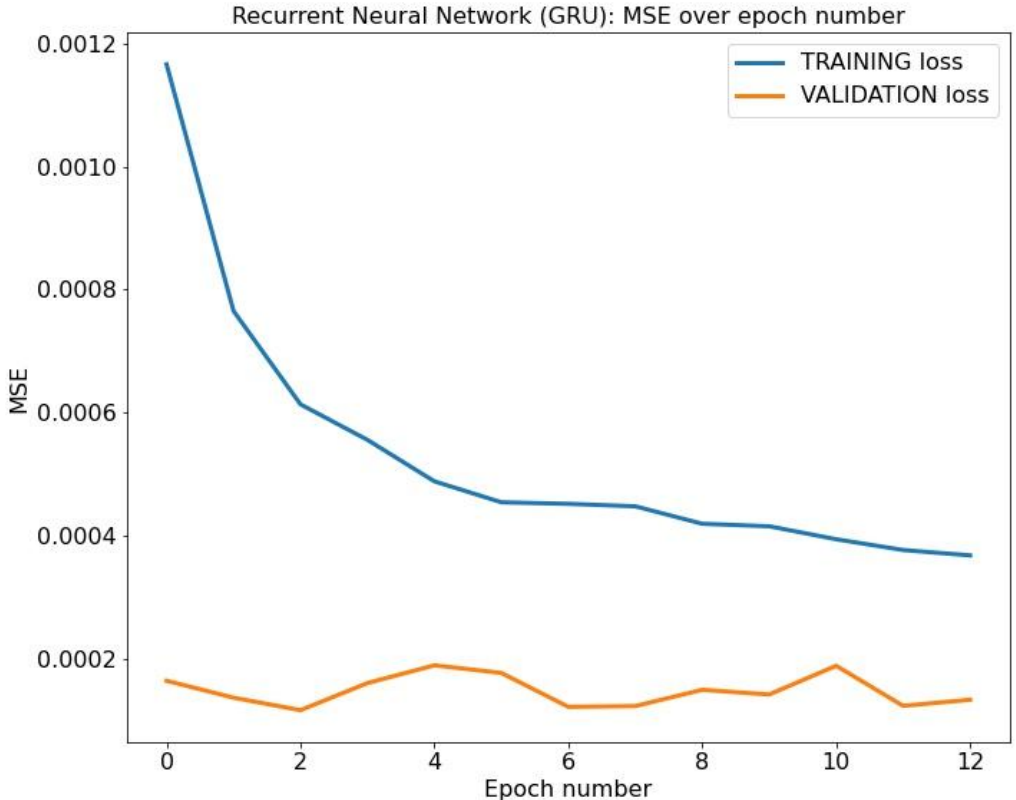
Recurrent Neural Networks

Layer	Units	# Parameters
GRU	100	30,900
Dropout (0.2)		
GRU	50	22,800
Dropout (0.2)		
Dense	10	510
Dense	1	11

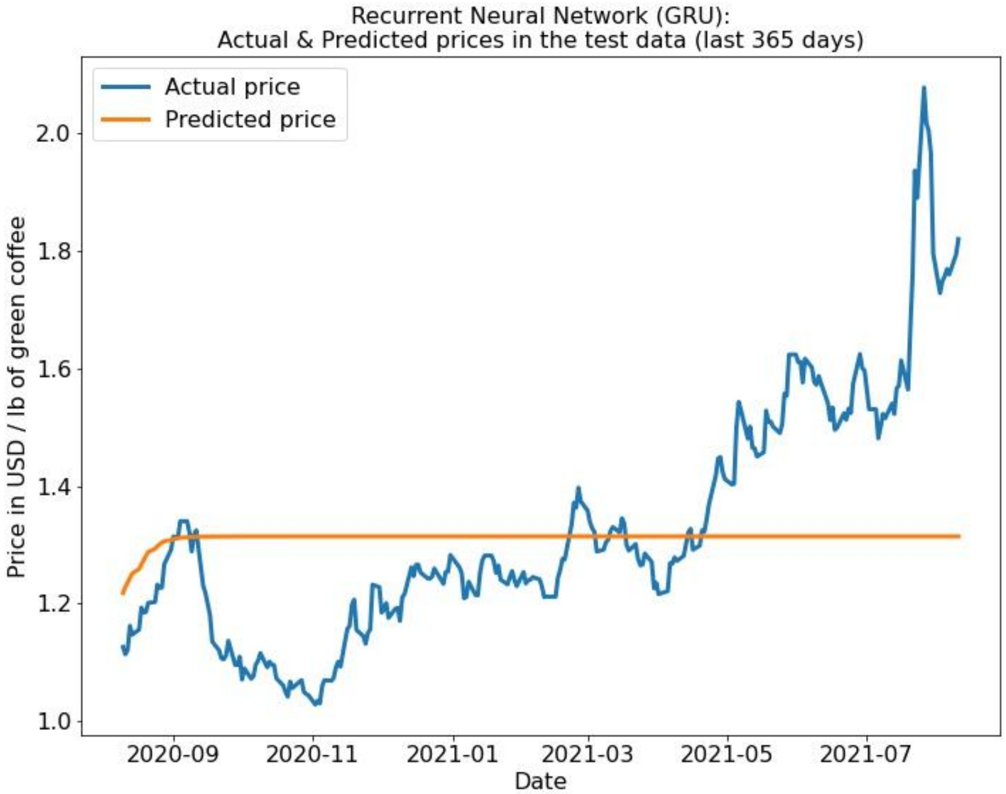
30 epochs

Including **early stopping**

RNN - Training and Validation Loss



RNN - Predictions on testing data



Testing MSE:

0.042

Testing MAPE:

11.38%

Summary & Conclusions

Models' metrics summary

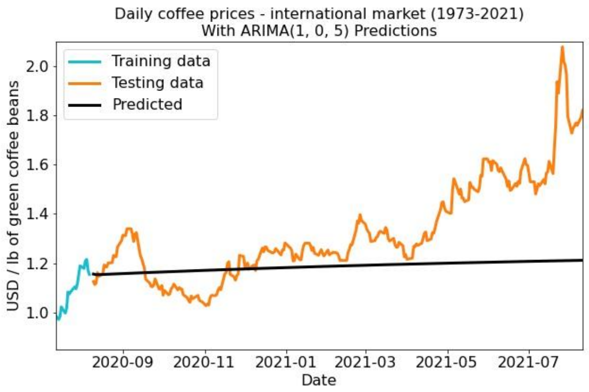
Model	Testing MSE	Testing MAPE
Baseline	1.585	115.73%
ARIMA	0.056	12.17%
Prophet	0.317	37.36%
RNN	0.042	11.38%

Models' metrics summary

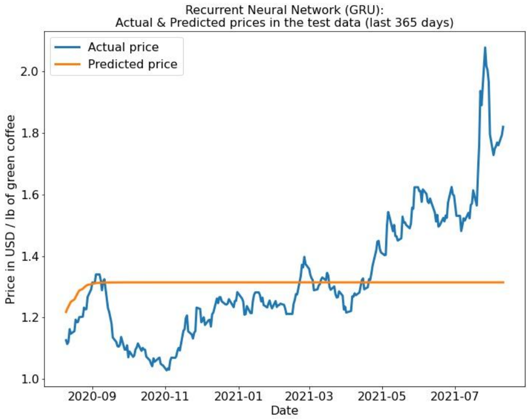
Model	Testing MSE	Testing MAPE
Baseline	1.585	115.73%
ARIMA	0.056	12.17%
Prophet	0.317	37.36%
RNN	0.042	11.38%

Conclusions

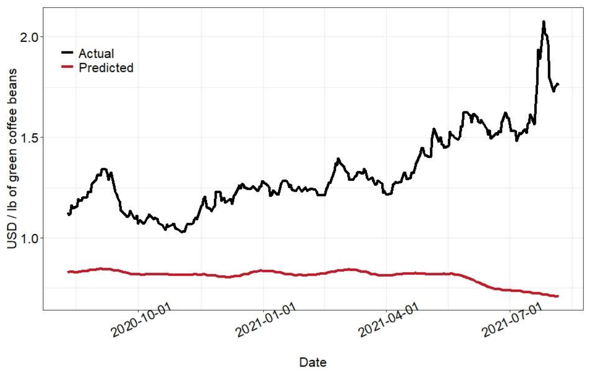
ARIMA



RNN



Prophet



Conclusions

- No clear pattern in the data - hard to forecast.

- Prophet did worst - due to lack of stable trend and seasonality.

- ARIMA and RNN - far from perfect, but significantly better.

Possible next steps

- **Improve RNN** performance.

- Integrate **additional predictors** (e.g., weather conditions; demand and supply; etc.)

Possible next steps

Until then… enjoy a 

/  of coffee!

(And don’t worry about how much it’ll cost tomorrow)

Thank you!