JPMC Quant Challenge

-Jayant Jain

Senior Undergraduate
Computer Science & Engineering
IIT Delhi

Outline

I’ll walk you through my approach and potential improvements to the following components of the challenge -

- Q1: Interpolating and Extrapolating the Natural Gas Price Data
- Q4: Evaluating the best contract given the future prices
- Q5: Price Simulation using the Geometric Brownian Motion Model

Part 4:

Contract Valuation given the Future Prices

Input

Conditions -

1. Fixed dates where Withdrawal/Injection is allowed from the storage facility
2. Fixed amount of Natural Gas that can be Withdrawn/Injected on each valid day
3. Cost to store the Natural Gas (cost per unit gas) and extra penalty if the maximum storage of the facility is exceeded
4. We can buy and sell multiple times.

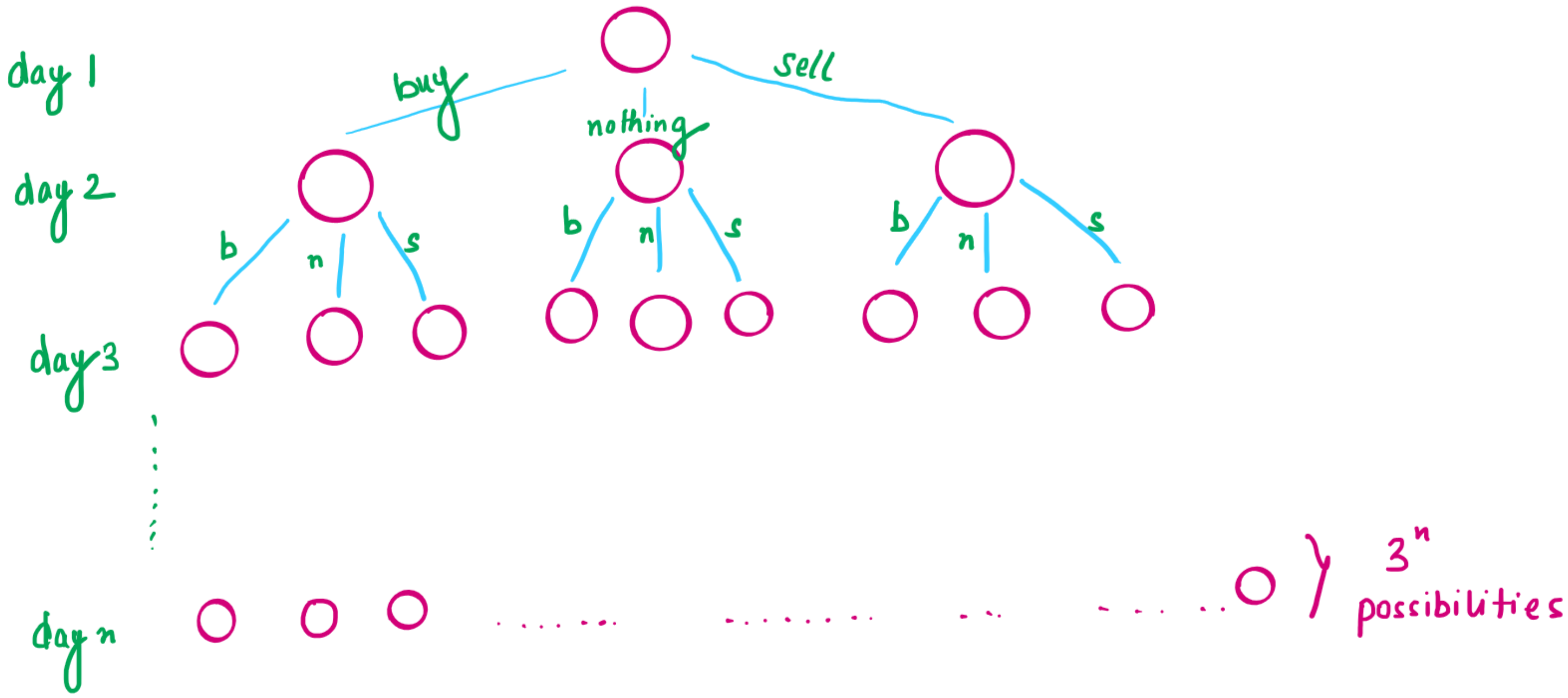
Naïve Approach

I considered all possible buying-selling patterns, and chose the one which gave the maximum profit.

For each day I have 3 options – **Buy** Natural Gas, **Sell** Natural Gas or do **Nothing**

For n number of valid dates, there are 3^n buying-selling patterns, and the evaluation of each pattern takes O(n) time.
Therefore order of this approach was O(n. 3^n)

This is Exponential in the number of days and does not scale well.



day 1
buy
nothing
sell
day 2
b
n
s
b
n
s
b
n
s
day3
day n
3^n possibilities

Optimisation

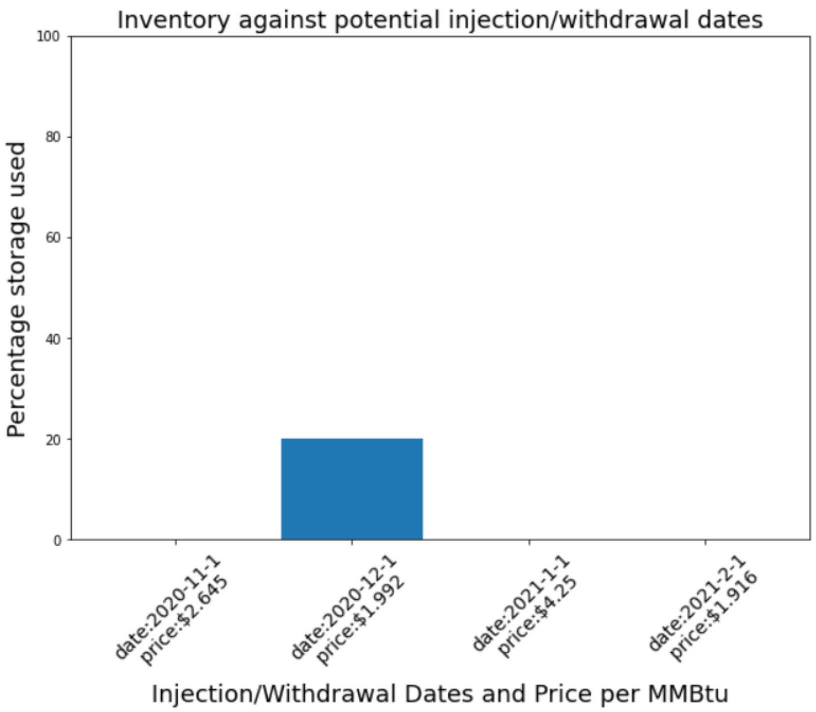
- We don’t need to evaluate from scratch at each leaf node.
- We can store the profit from our decisions at each node and that helps make the algo become O(3^n) instead of O(n. 3^n)

- Another optimisation is to stop expansion when we hit an invalid node.
- For example I can't sell more than I have already bought and such patterns can be ignored/avoided early on.

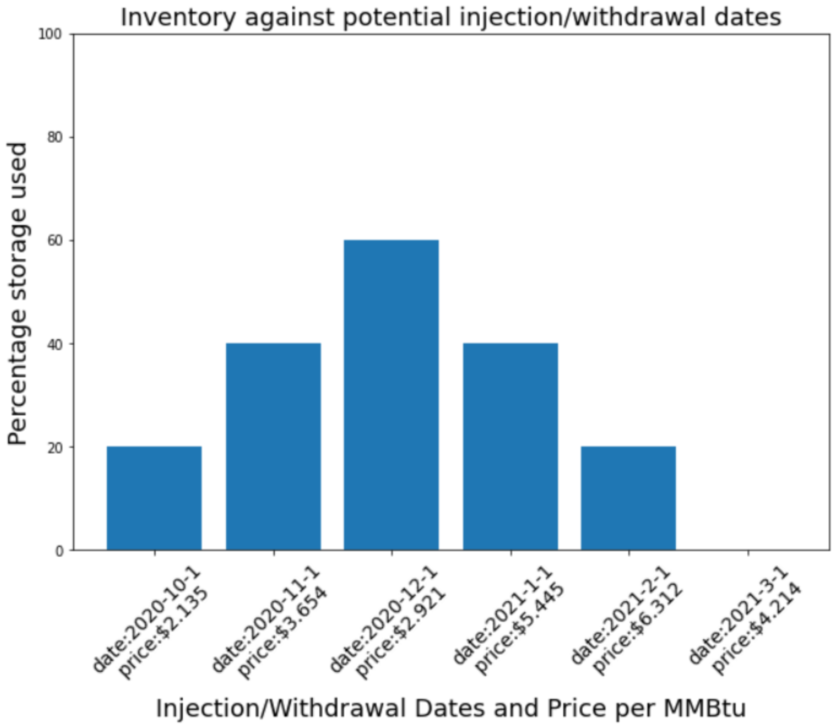
Correctness

- Heavy on computation but gives the exact answer
- My answer for the given testcase is $13.45128 which matches the given answer

Volumes Profile

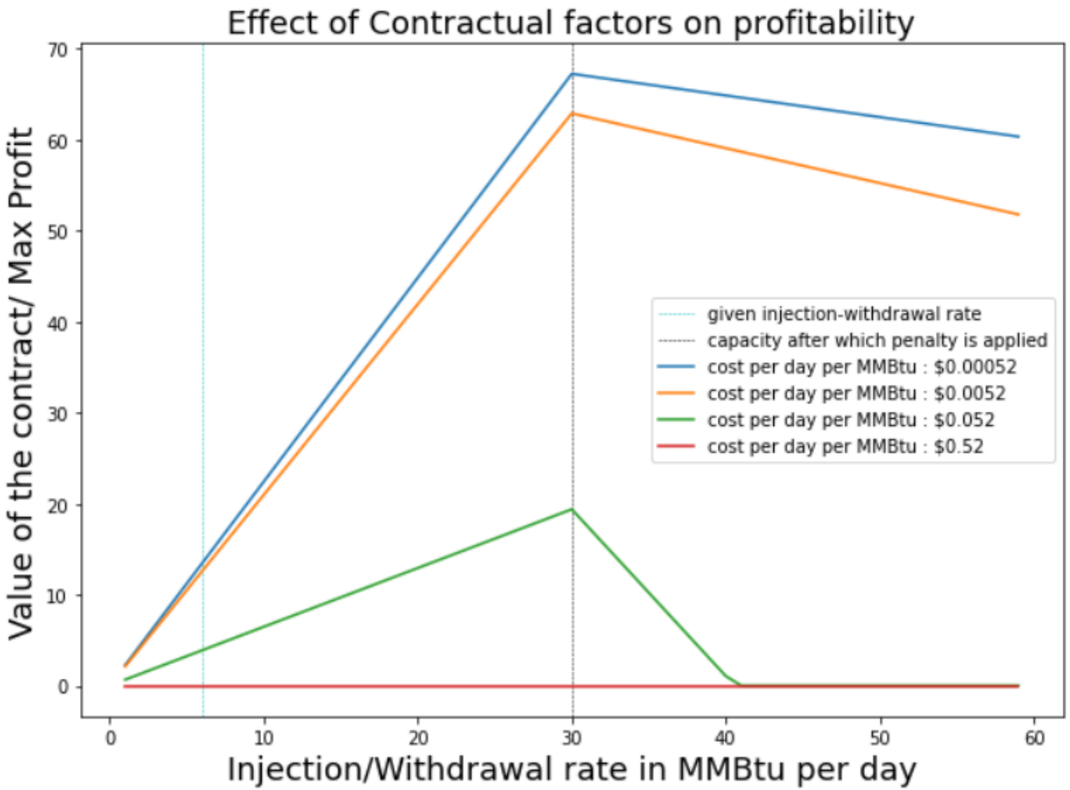


For the Given testcase



For the default testcase during the competition

Effect of Contractual Factors



Part 5:

Contract Valuation by estimating the Future Prices

Problem Statement

Given: Initial price and the Stochastic Differential Equation that governs it's evolution

Aim: To find the value of the contract

Stages

1. Find the expectation of price on each potential Injection/Withdrawal date (these dates are given)
2. Assuming these prices are correct, find the maximum profit

Simulating the price evolution

Geometric Brownian Motion

$$\frac{dS_t}{S_t} = \mu_t dt + \sigma dw$$

$$dw \sim \mathcal{N}(0, dt)$$

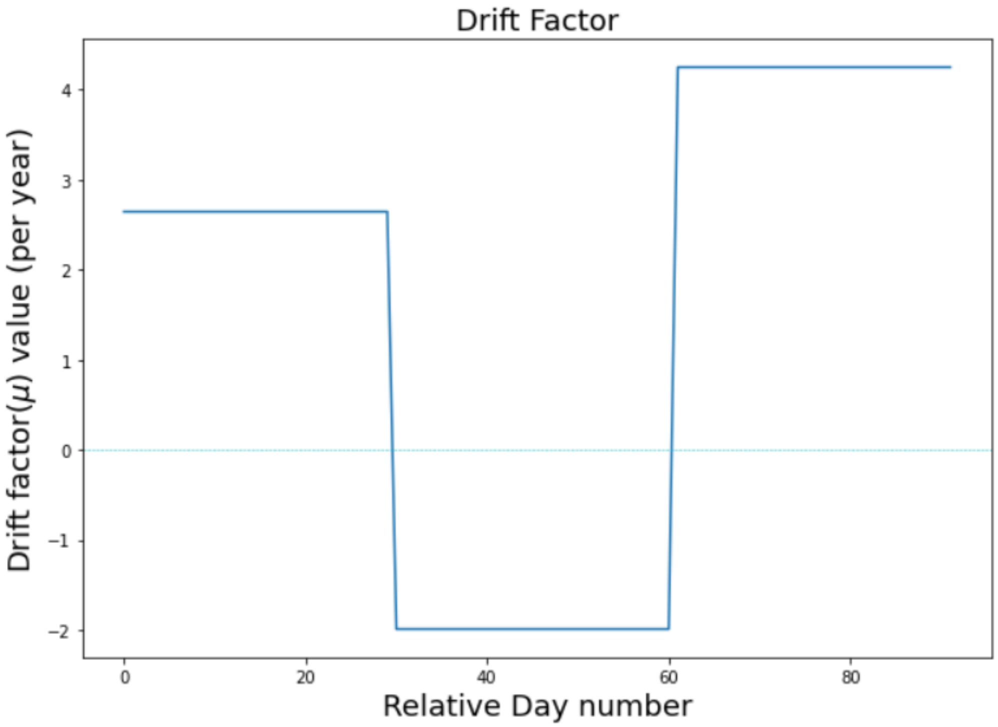
using Ito's Lemma

$$S_T = S_0 \cdot exp((\int_0^T \mu_t dt) - \frac{\sigma^2}{2}T + \sigma W_T)$$

$$W_T \sim \mathcal{N}(0, T)$$

$$E(S_T) = S_0 \cdot exp(\int_0^T \mu_t dt)$$

Variable Drift Factor

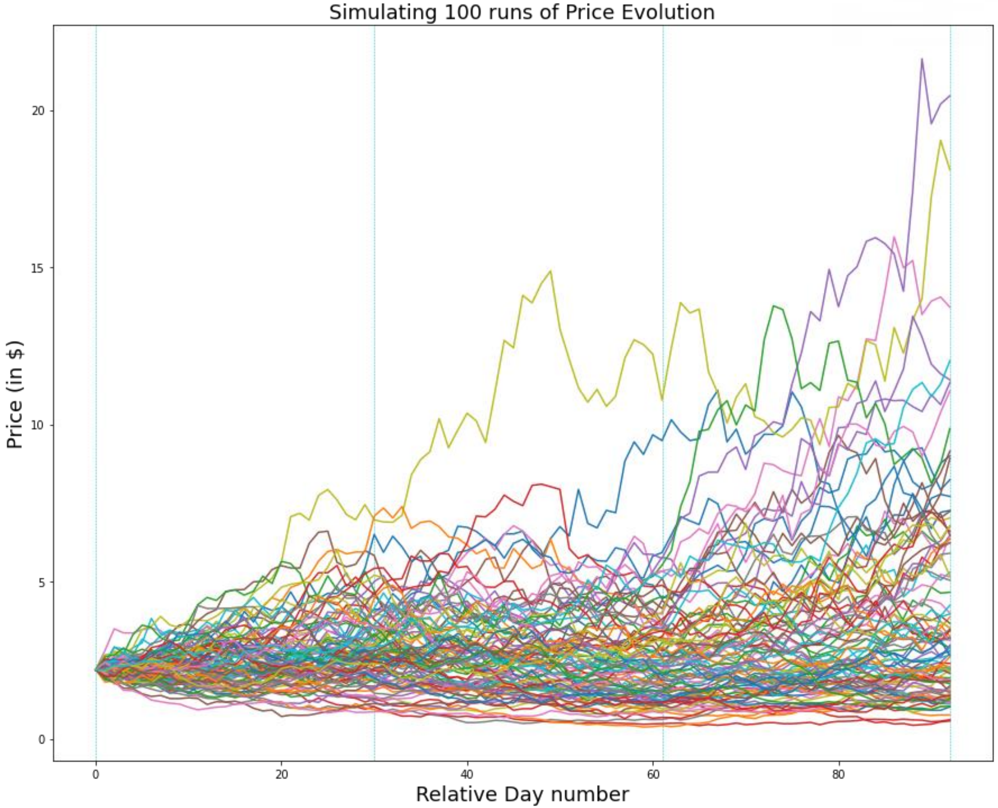


Simulating the price evolution

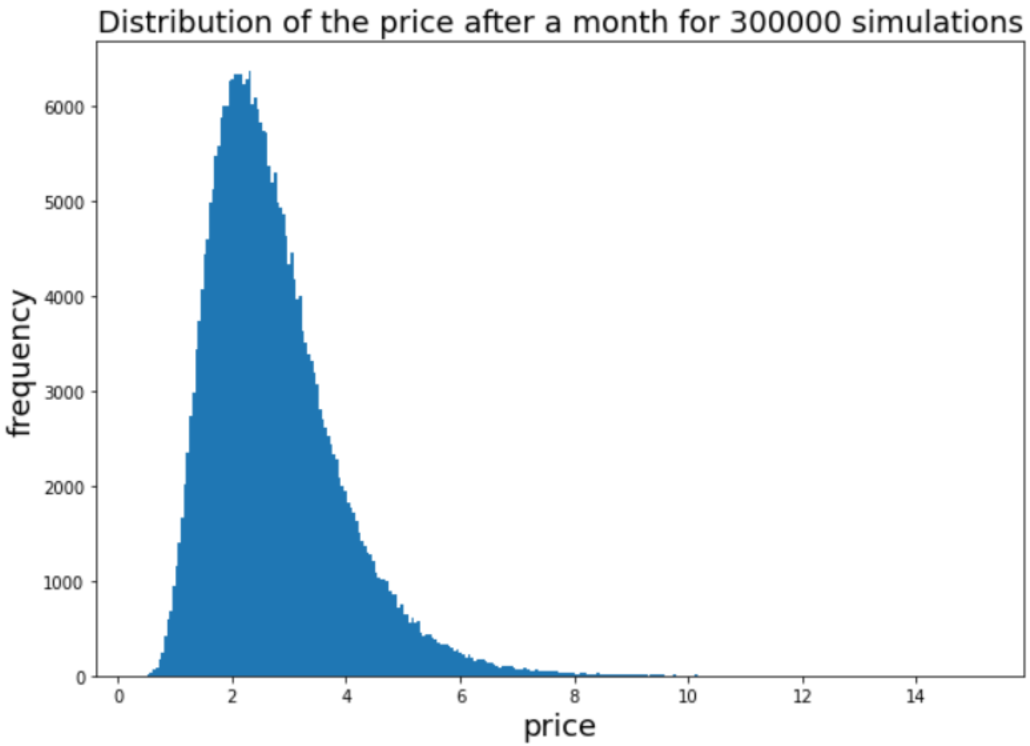
- Taking day-sized steps

$$S_t = S_{t-\Delta t} \cdot exp((\mu_{t-\Delta t} - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}z)$$

$$z \sim \mathcal{N}(0,1)$$



Distribution of the price after a month



A shortcut?

- Using 30,000 Paths,
  - Time consuming
  - Some amount of error because we can't do infinite trials
  - Prices came out to be - `[2.2, 2.728044, 2.303834, 3.291503] (in dollars)`
- Directly using the expectation,
  - Prices (using the closed expression) = `[2.2, 2.734243, 2.308669, 3.312257]`

Contract Valuation given the prices

- Here the conditions are a bit different
  - We can only buy and sell once
  - There is no "Rate" of withdrawal / injection

Contract Valuation given the prices

We have to find d1 and d2 such that

$$\underbrace{(\text{price}[d2] - \text{price}[d1])}_{\text{(selling - buying)}}*\text{capacity\_of\_storage\_facility} - \text{daily\_cost\_of\_storage}*\underbrace{(d2 - d1)}_{\text{no. of days stored}}$$

is maximised,
where d1, d2 belong to the given injection/withdrawal dates and d2 > d1

We can transform the array of prices (by taking differences of consecutive elements and including the index) and then apply Kadane's Algorithm to find the maximum subarray which would give us the answer in O(n) time.

$$\text{transformed}[i] = (\text{prices}[i+1] - \text{prices}[i])*\text{capacity} – \text{daily\_cost\_of\_storage}$$

My answer for the given testcase is : `35.592095 Million Dollars`

Part 1 :

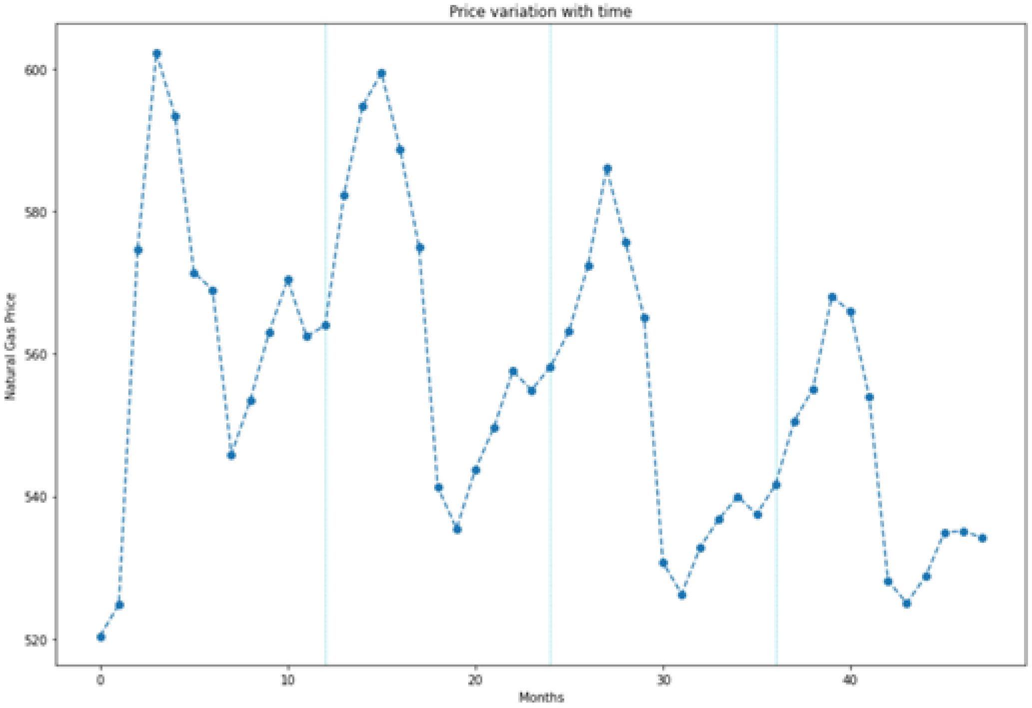
Interpolation & Extrapolation of Natural Gas Price Data

Aim

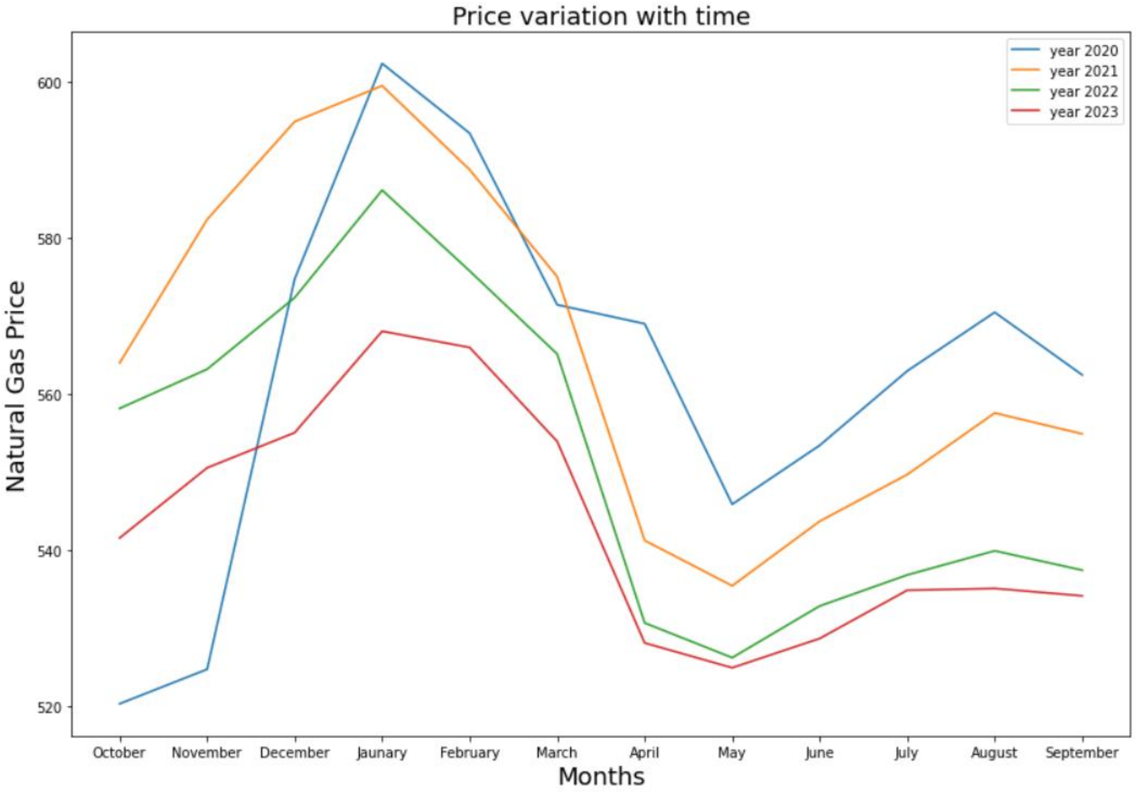
Given : Prices of Natural Gas at the end of each month for 4 years (October 2020 to September 2024)

To Do : Estimate Prices for the

- missing dates in between
- future dates as well



Seasonal Nature of the Prices



Interpolation

Simple linear interpolation for estimating prices on missing dates

For example, say the query date is 20th December, 2021.

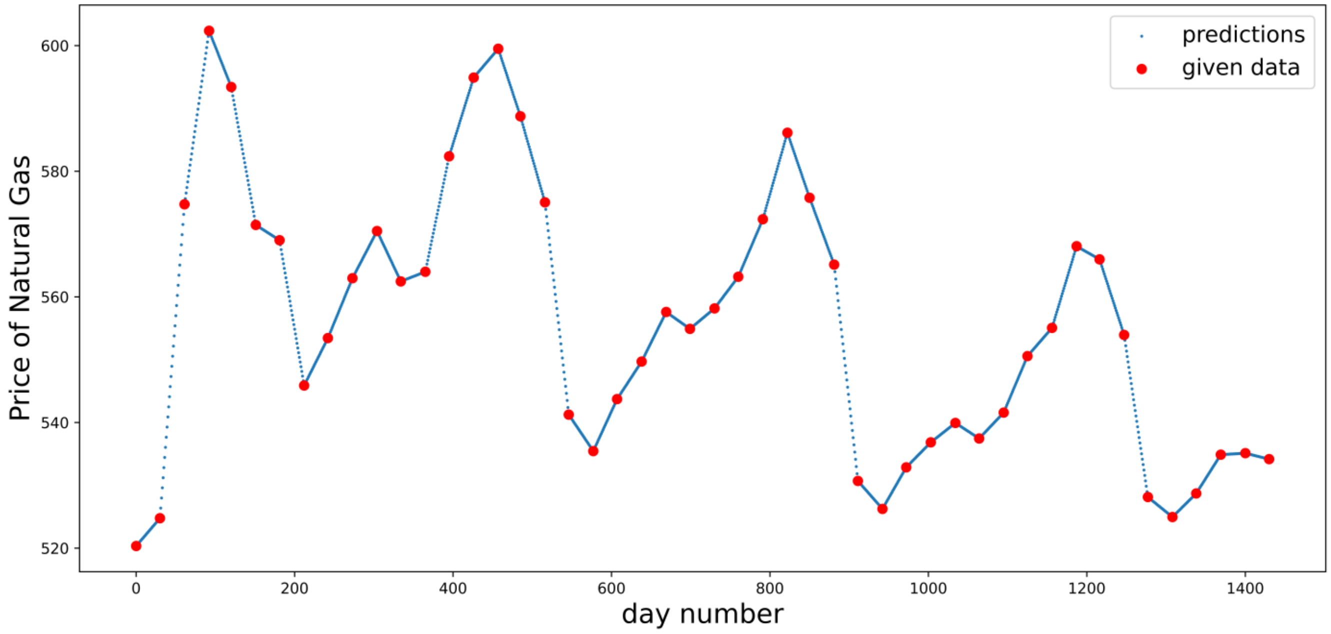
Right neighbour is 31st December 2021
Left neighbour is 30th November 2021

p1 = price on the left date
p2 = price on the right date

w1 = number of days between the left and query date = 20
w2 = number of days between the right and query date = 11

Prediction = (w1*p2 + w2*p1)/(w1 + w2)

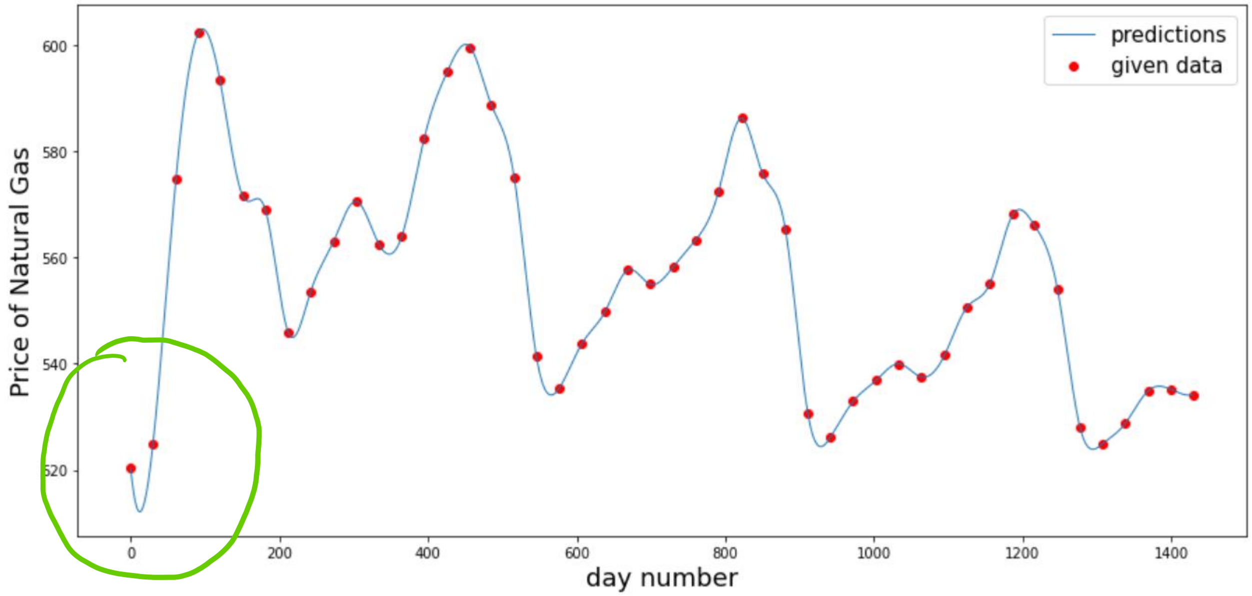
Interpolation : Linear



Drawbacks

- We have 48 known prices (4years, 12 months) BUT for interpolation we use only 2 neighbours
- Instead of making a piecewise linear function we can find a smoother fit – for example using a cubic spline

may not be a good prediction



Extrapolation

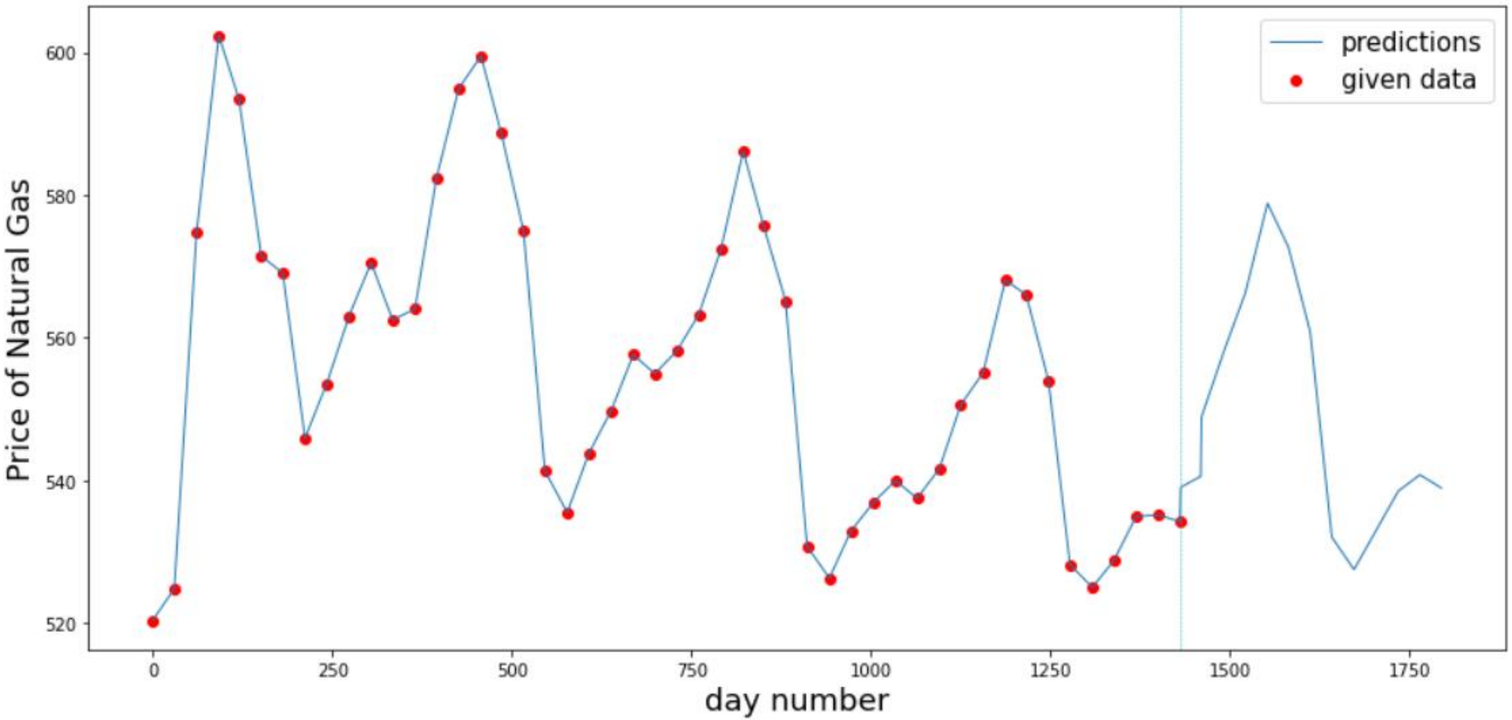
Goal : To make predictions for October 2024 – September 2025, given the data of the previous 4 years

An example,
Say the query date is 8th April 2025. I use the prices on 31st March and 31st April for all the previous years, that is, 8 data points

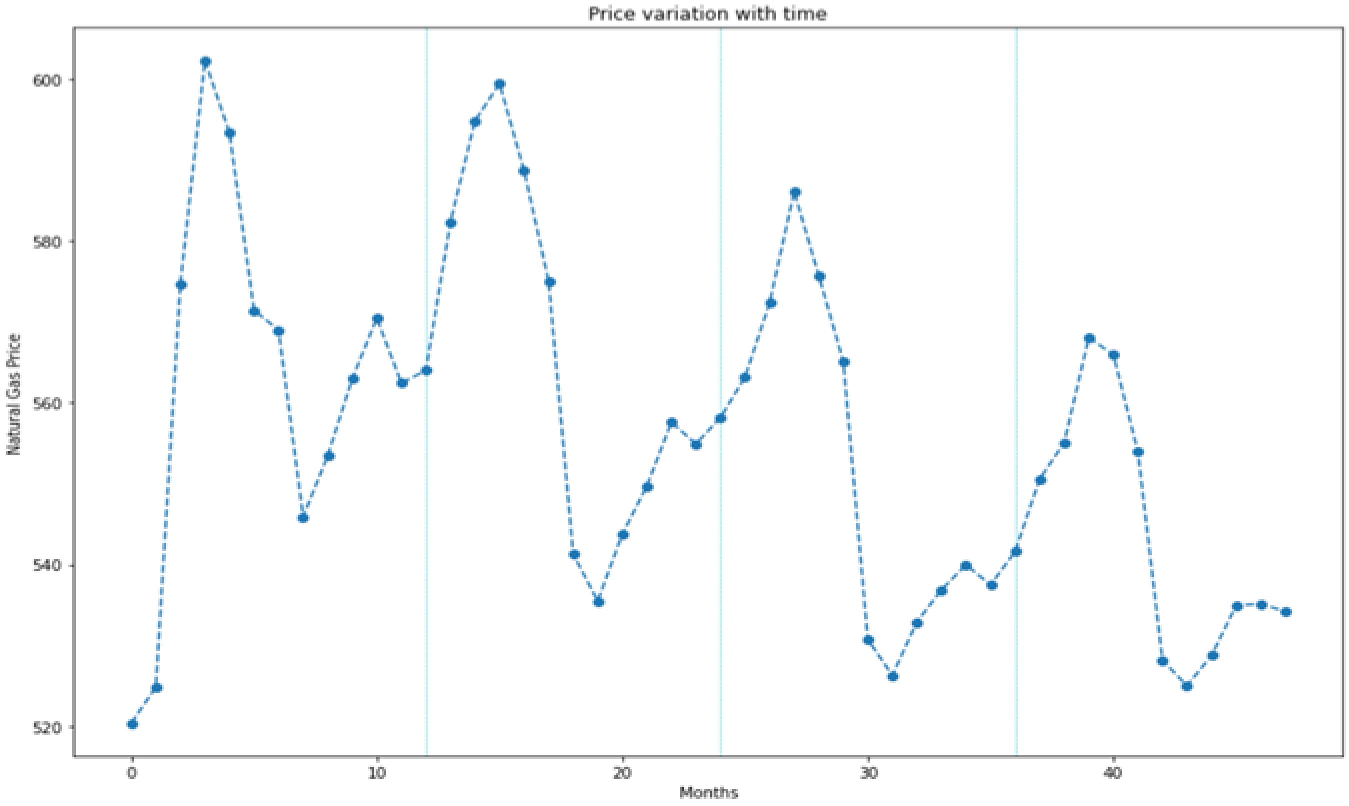
I take the weighted average of the predictions that my linear interpolator would have made for 8th April 2021, 8th April 2022, 8th April 2023 and 8th April 2024.

I weigh the more recent years(2024) more than the older ones (2021).

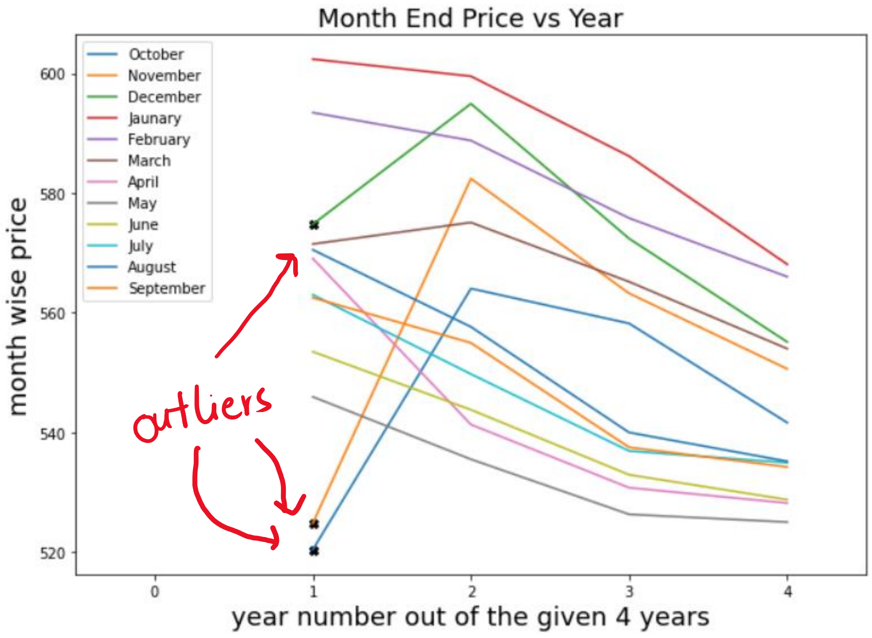
Extrapolation: Plot

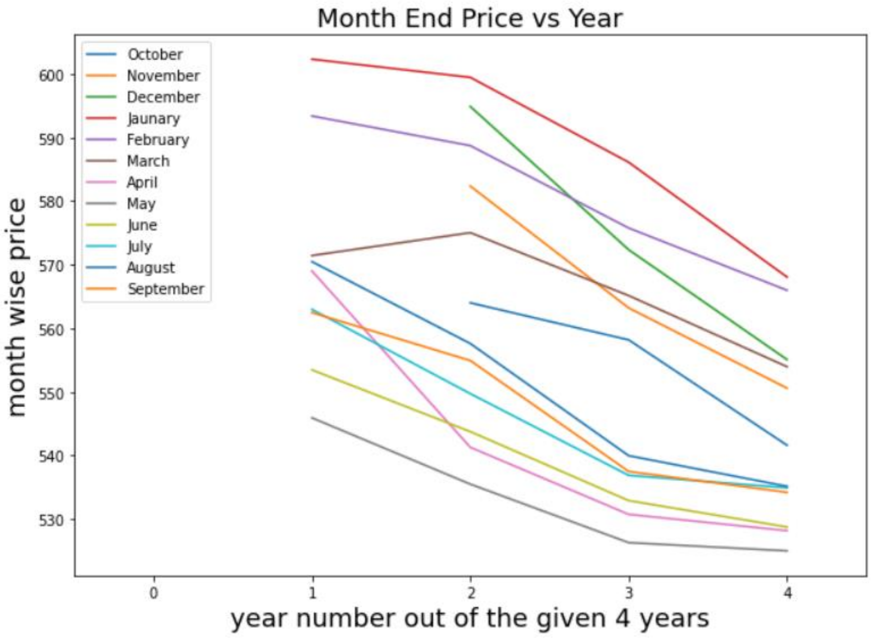


An observation: General Decrease in Overall Price



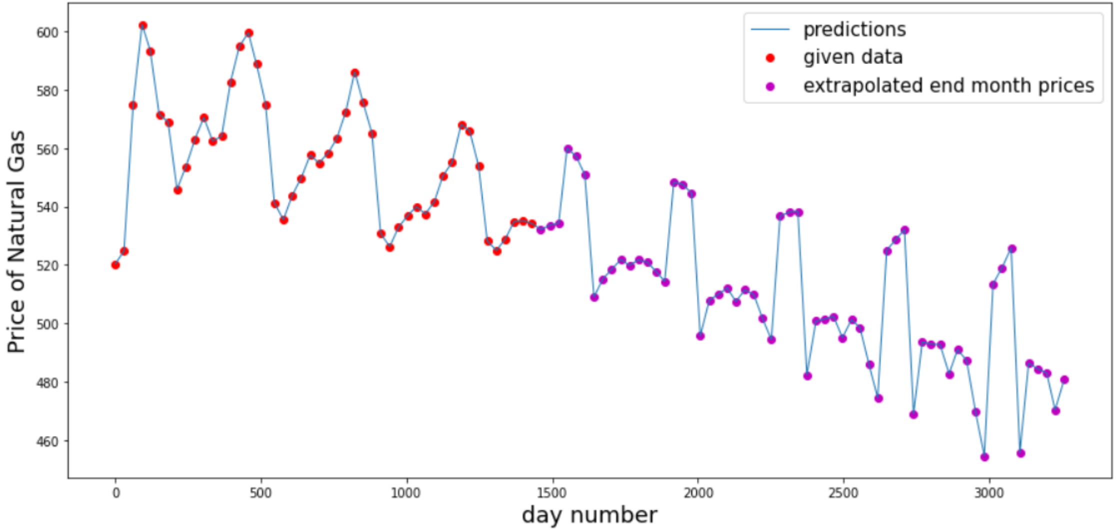
An Observation : General Decrease in Overall Price





An Observation : General Decrease in Overall Price

- We can fit a line (Linear Regression) through the month wise(end date of each month) prices to get the end month price for future years
- In between dates can further be found using linear interpolants
- For the next 5 years the plot would look like



Thank You