2Cents Capital - Derivatives Analyst Intern Task - Rishabh Mishra

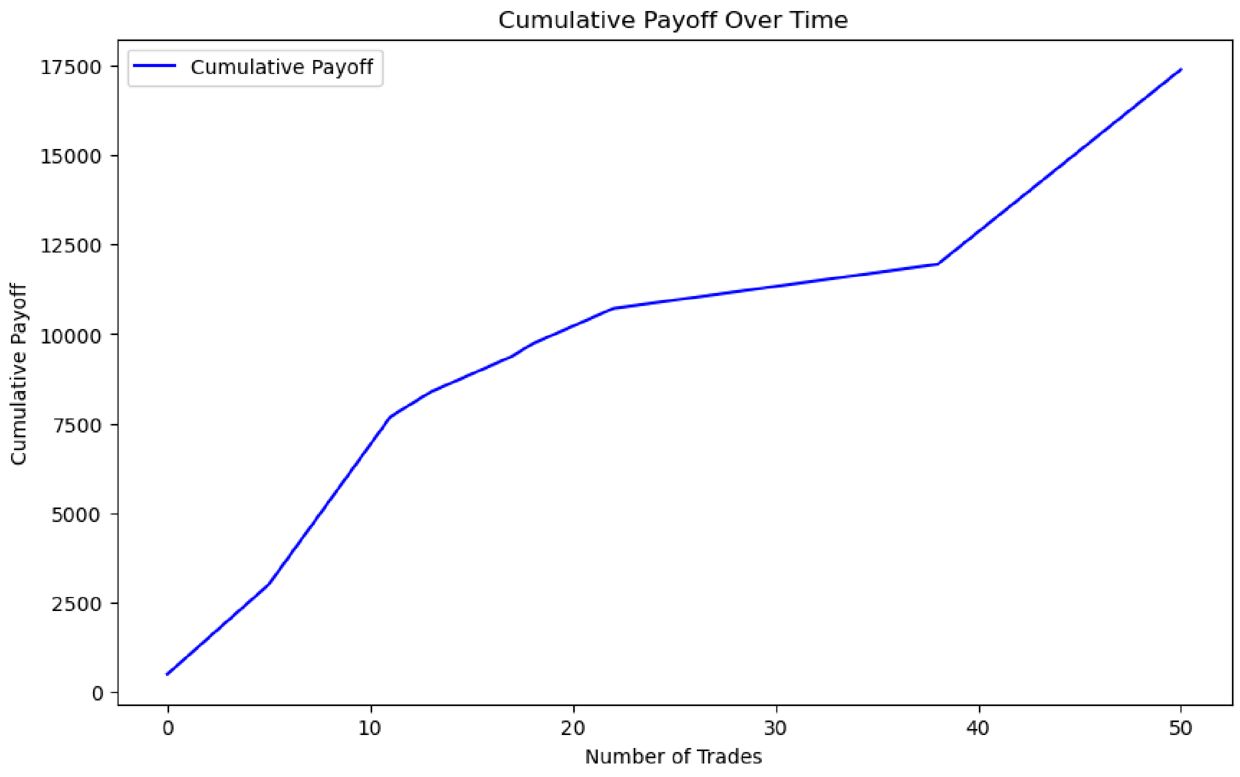
Task-1

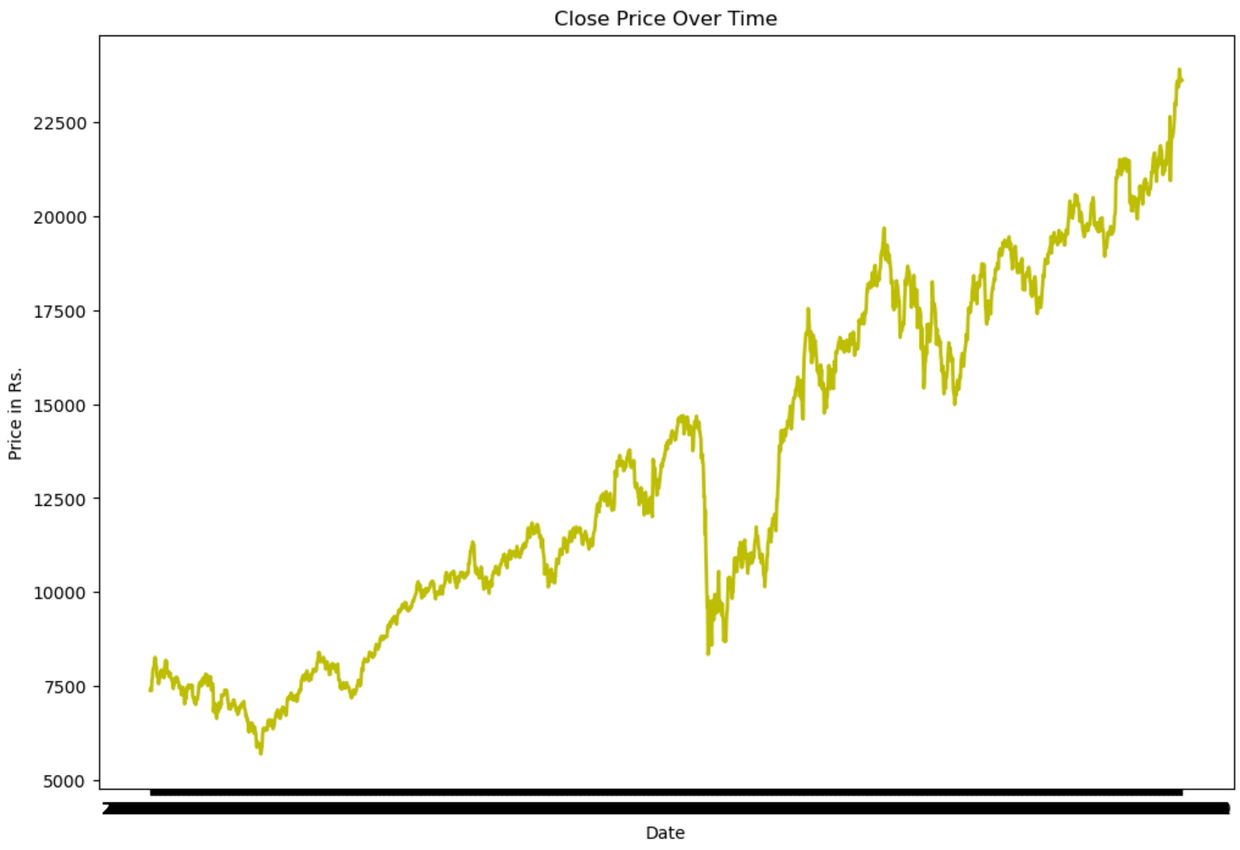
Scenario of a High Volatility market with high implied volatility

Due to a high volatility market, we can profit using a long straddle wherein we buy a call and put option which is at-the-money, in this case, we are using a european option which means that the option can't be exercised early.

We use the FIN-NIFTY index since the NIFTY index option files weren't being parsed properly (throwing an error)

1. Merge all the options information into one csv to work with.
2. Apply a filter for implied volatility greater than 0.8
3. For a straddle, the expiry date needs to be the same, so a new column was created for the expiry date to be matched further
4. Now, an At-the-money option is not realistic, so a 5% margin for error was given on the spot to match for an ATM option with respect to the strike price of the call and put
5. Created a straddle where the expiry date and the strike price of the call and put are the same which gave us our required trades
6. Now, going through the minute by minute FIN-NIFTY index prices, converted the dataset over to a day-by-day price dataset, used the close prices of the days to calculate the payoff of the straddle at expiry

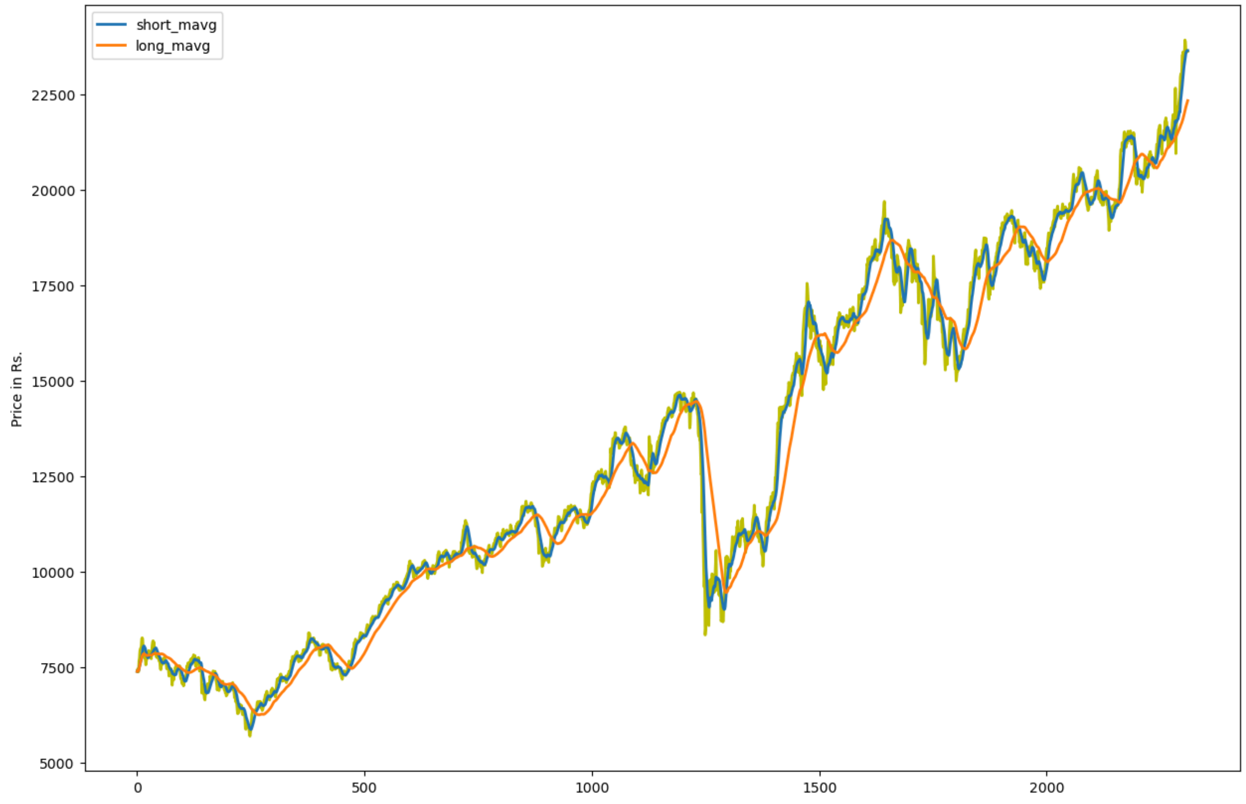

Cumulative Payoff Over Time
Cumulative Payoff
17500
15000
12500
10000
7500
5000
2500
0
0
10
20
30
40
50
Number of Trades
Cumulative Payoff


Close Price Over Time
22500
20000
17500
15000
12500
10000
7500
5000
Price in Rs.
Date

FIN-NIFTY index over time (2015-2024)

Bullish Market with Low Volatility

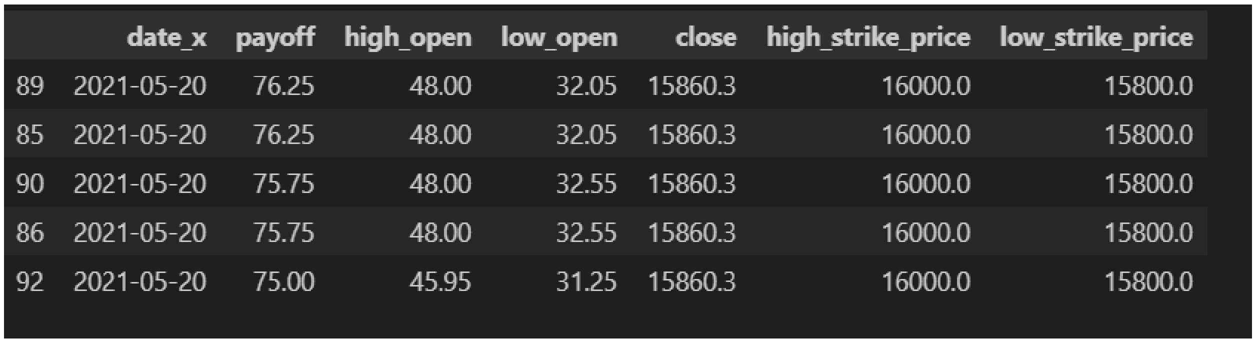
The FIN-NIFTY index has been used here as well. In order to find the entry points to find the bullish segments of a market, a 10-day MA and a 50-day MA was calculated and plotted as follows,



When the short duration MA crosses the long duration MA, it signals a bullish trend and a position is taken on that day (for the sake of simplicity), we take a bull call spread (where a call option is bought at a strike price lower than the current spot price and a call option is shorted at a strike price higher than current spot price)

Similar to the above strategy, the payoff is calculated at expiration after matching with the FIN-NIFTY index close price of the expiry date, this gives us the respective payoff

The payoff includes the selling and buying prices of the option.



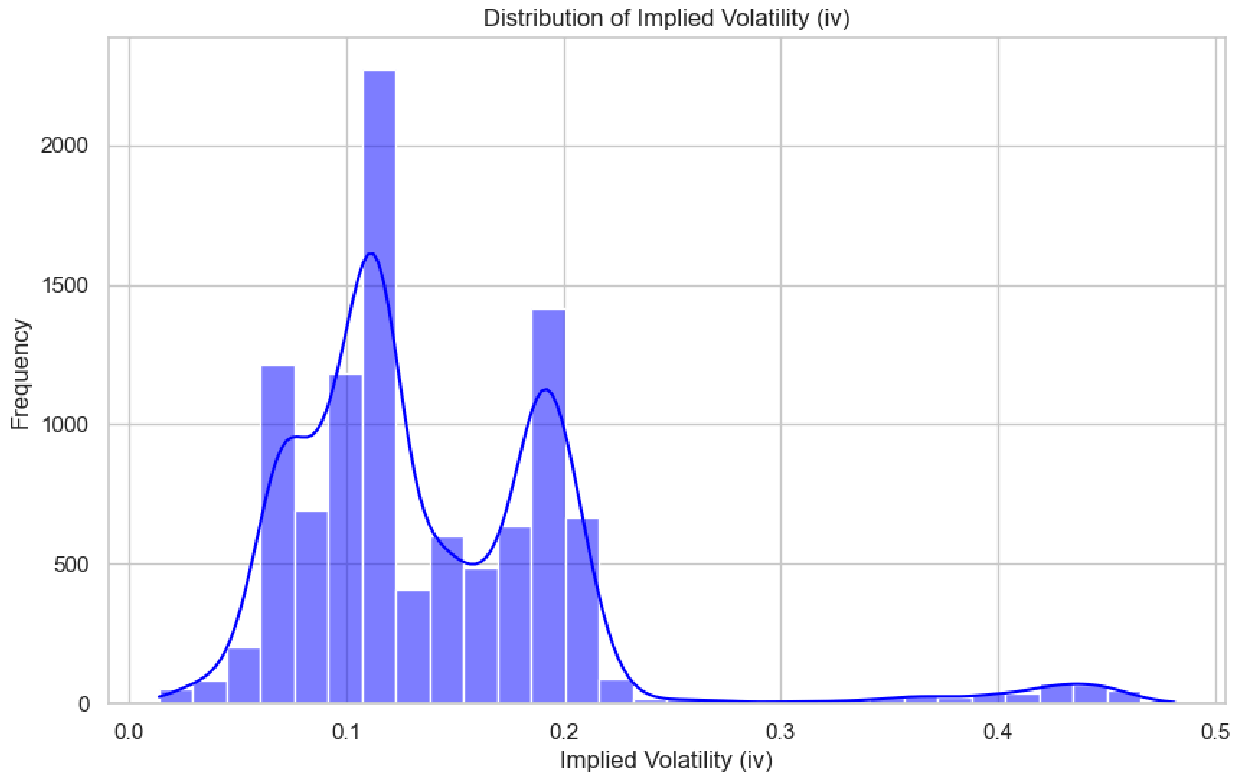
	date_x	payoff	high_open	low_open	close	high_strike_price	low_strike_price
89	2021-05-20	76.25	48.00	32.05	15860.3	16000.0	15800.0
85	2021-05-20	76.25	48.00	32.05	15860.3	16000.0	15800.0
90	2021-05-20	75.75	48.00	32.55	15860.3	16000.0	15800.0
86	2021-05-20	75.75	48.00	32.55	15860.3	16000.0	15800.0
92	2021-05-20	75.00	45.95	31.25	15860.3	16000.0	15800.0

As we can see, the payoff is capped off at a threshold which signifies the maximum profit of the respective bull call spread.

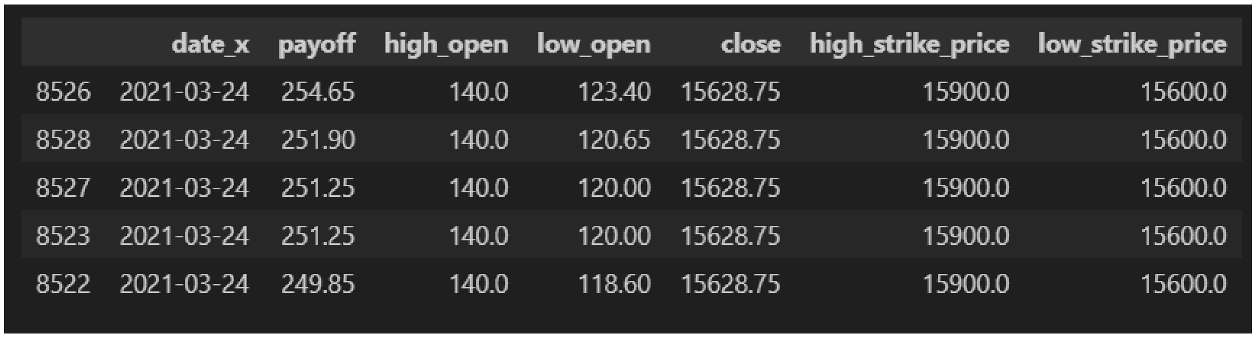
Task-1

Bearish Market with high volatility

Reversing the above strategy, a bearish position was taken by using a bear put spread on the same index, this gave the below results



But the volatility of the call options was really low, in the bearish trends.



	date_x	payoff	high_open	low_open	close	high_strike_price	low_strike_price
8526	2021-03-24	254.65	140.0	123.40	15628.75	15900.0	15600.0
8528	2021-03-24	251.90	140.0	120.65	15628.75	15900.0	15600.0
8527	2021-03-24	251.25	140.0	120.00	15628.75	15900.0	15600.0
8523	2021-03-24	251.25	140.0	120.00	15628.75	15900.0	15600.0
8522	2021-03-24	249.85	140.0	118.60	15628.75	15900.0	15600.0

Task-2 Analysis of Calendar Call Spreads

- Short call with 60 days to expiry and long call with 70 days to expiry.

There was no dataset which had the same expiry date and 60 and 70 days to expiry respectively.

- Short call with 0 days to expiry and long call with 1 day to expiry.

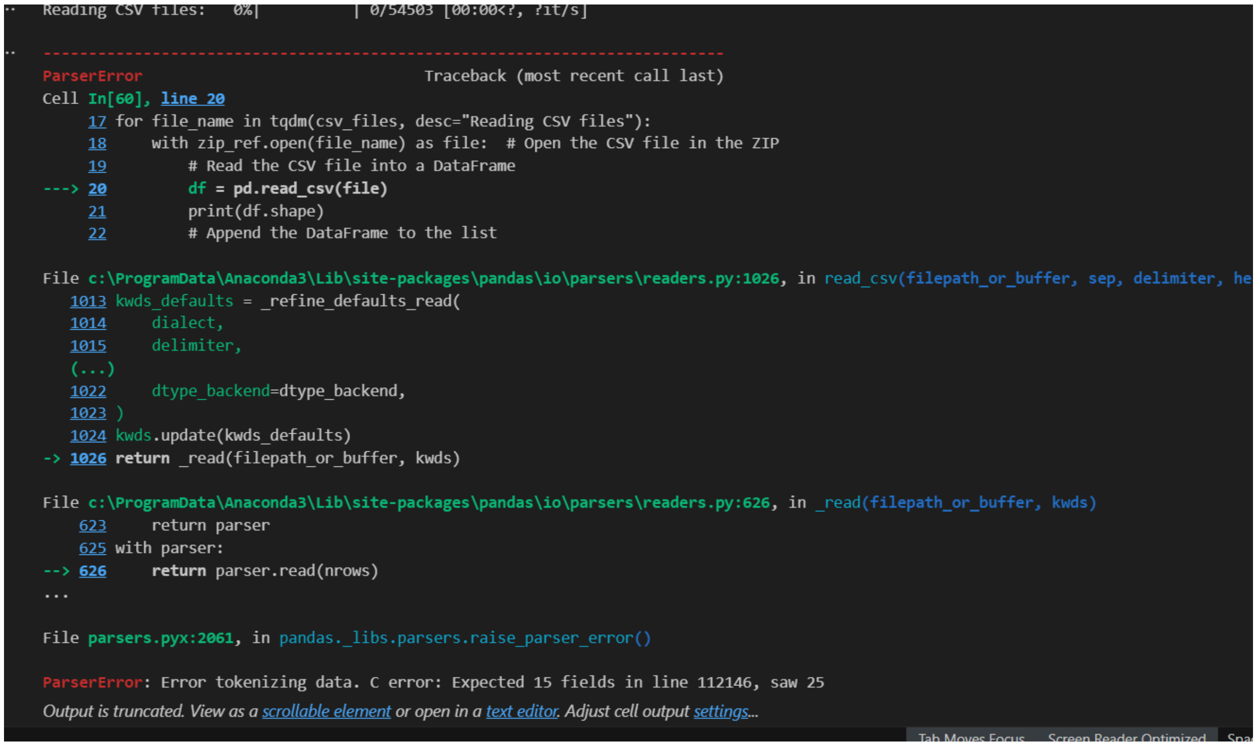
The dataset was too large to handle ans was crashing the kernel again and again

The code wasn't working as expected and there was some issue when I tried to make it work on a part of the dataset.

Task-3

The best strategy would be found on the basis of the Sharpe Ratio and Sortino Ratio.

Error being thrown trying to parse nifty files



```
Reading CSV files:   0%|          | 0/54503 [00:00<?, ?it/s]

---------------------------------------------------------------------------
ParserError                               Traceback (most recent call last)
Cell In[60], line 20
     17 for file_name in tqdm(csv_files, desc="Reading CSV files"):
     18     with zip_ref.open(file_name) as file:  # Open the CSV file in the ZIP
     19         # Read the CSV file into a DataFrame
---> 20         df = pd.read_csv(file)
     21         print(df.shape)
     22         # Append the DataFrame to the list

File c:\ProgramData\Anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, he
   1013 kwds_defaults = _refine_defaults_read(
   1014     dialect,
   1015     delimiter,
   (...)
   1022     dtype_backend=dtype_backend,
   1023 )
   1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File c:\ProgramData\Anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:626, in _read(filepath_or_buffer, kwds)
    623     return parser
    625 with parser:
--> 626     return parser.read(nrows)
...

File parsers.pyx:2061, in pandas._libs.parsers.raise_parser_error()

ParserError: Error tokenizing data. C error: Expected 15 fields in line 112146, saw 25
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...