Trading Bot using Donchian Channels Strategy

Importing the Data and Indicators

```
[106] result = yf.download('^NSEI', start = "2000-01-01")

      [**********************100%**********************]  1 of 1 completed
```

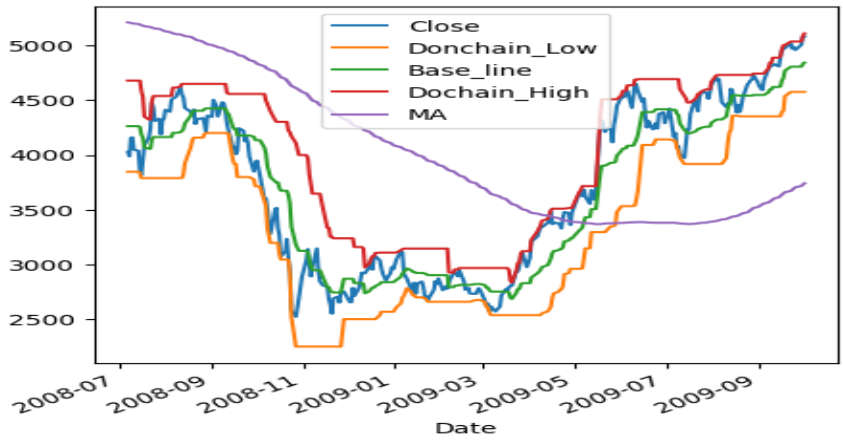
```
[107] def DONCHIAN(hi, lo, n):
          hi = pd.Series(hi)
          lo = pd.Series(lo)
          uc = hi.rolling(n, min_periods=n).max()
          lc = lo.rolling(n, min_periods=n).min()
          mc = (uc + lc) / 2
          return lc, mc, uc
      tf = DONCHIAN(result.High, result.Low, 20)
      nf = pd.DataFrame(tf)
      nf = nf.T
      nf = nf.rename(columns={'Low':'Donchain_Low','High':'Dochain_High', 'Unnamed 0':'Base_line'})
      result = result.Close.to_frame()
      result = pd.concat([result, nf.reindex(result.index)], axis=1)
      result['MA'] = result.Close.rolling(200).mean()
      result = result.dropna()
```

```
[108] result
```

Date	Close	Donchain_Low	Base_line	Dochain_High	MA
2008-07-07	4030.000000	3848.250000	4264.000000	4679.750000	5210.716744
2008-07-08	3988.550049	3848.250000	4264.000000	4679.750000	5208.186245
2008-07-09	4157.100098	3848.250000	4264.000000	4679.750000	5206.240745
2008-07-10	4162.200195	3848.250000	4264.000000	4679.750000	5203.389995
2008-07-11	4049.000000	3848.250000	4264.000000	4679.750000	5199.897246
...	...	...	...	...	...
2023-04-06	17599.150391	16828.349609	17300.349609	17772.349609	17511.112983
2023-04-10	17624.050781	16828.349609	17300.349609	17772.349609	17522.765737
2023-04-11	17722.300781	16828.349609	17288.549805	17748.750000	17534.626489
2023-04-12	17812.400391	16828.349609	17327.049805	17825.750000	17545.494492
2023-04-13	17828.000000	16828.349609	17335.250000	17842.150391	17557.567993

3617 rows × 5 columns

Plotting the Close Price with Indicators



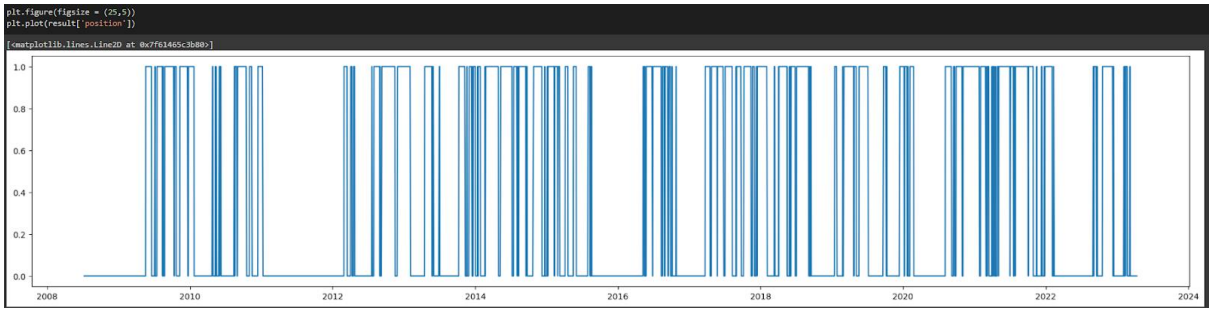
Creating Buy and Sell Signals

```python
# BUY condition
result['signal'] = np.where((result['Close'] > result['MA']) & (result['Close'] > result['Base_line']) & (result['Close'].shift(1) <= result['Base_line']),1,0)
# SELL condition
result['signal'] = np.where((result['Close'] < result['MA']) | ((result['Close'] < result['Base_line']) & (result['Close'].shift(1) >= result['Base_line'])),-1 ,result['signal'])
# creating long and short positions
result['position'] = result['signal'].replace(to_replace=0, method='ffill')
# shifting by 1, to account of close price return calculations
result['position'] = result['position'].shift(1)
result['position'] = result['position'].replace(-1, 0)
# calculating buy and hold strategy returns
result['bnh_returns'] = np.log(result['Close']/result['Close'].shift(1))
# calculating stretegy returns
result['strategy_returns'] = result['bnh_returns'] * (result['position'])
```

Plotting the Positions

```python
plt.figure(figsize = (25,5))
plt.plot(result['position'])
```

```
[<matplotlib.lines.Line2D at 0x7f61465c3b80>]
```



Backtesting and Plotting the Results