BUILDING MOVING AVERAGE TRADING STRATIGY 📊

For apple stock

In [1]:

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import seaborn as sns
plt.style.use('seaborn-v0_8')
```

Downloading stock data using yahoo finanace

In [2]:

```
ticker = 'AAPL'
start_date = '2020-01-01'
end_date = '2024-12-31'

data = yf.download(ticker, start=start_date, end=end_date)
data.head()
```

```
YF.download() has changed argument auto_adjust default to True
[*********************100%***********************]  1 of 1 completed
```

Out[2]:

Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2020-01-02	72.620842	72.681289	71.373218	71.627092	135480400
2020-01-03	71.914825	72.676454	71.689965	71.847125	146322800
2020-01-06	72.487831	72.526518	70.783234	71.034694	118387200
2020-01-07	72.146935	72.753816	71.926907	72.497522	108872000
2020-01-08	73.307518	73.609752	71.849540	71.849540	132079200

Calculate Moving Averages 🧮

In [3]:

```
# Calculate 50-day and 200-day SMA
data['SMA50'] = data['Close'].rolling(window=50).mean()
data['SMA200'] = data['Close'].rolling(window=200).mean()
```

In [4]:

```
# Create signal: 1 = Buy, -1 = Sell
data['Signal'] = 0
data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1, -1)

# Position: when to hold the stock
data['Position'] = data['Signal'].shift(1)
```

```
C:\Users\Prajwal\AppData\Local\Temp\ipykernel_18240\2397333694.py:3: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1, -1)
```

In [ ]:

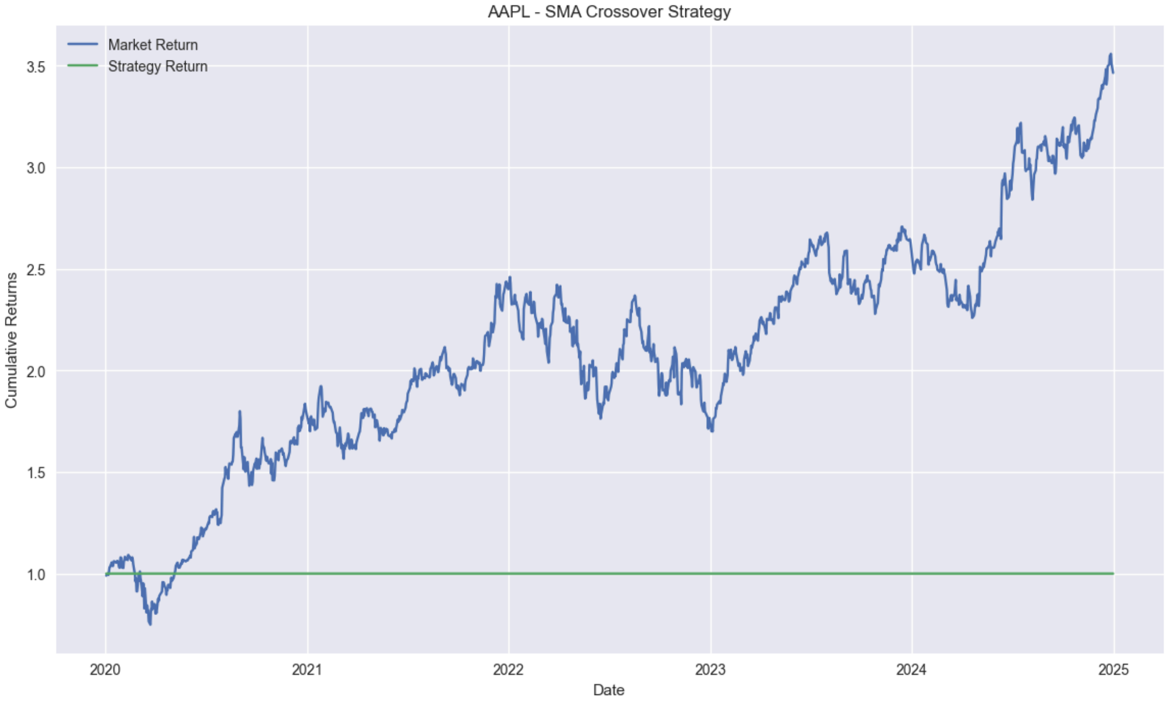
In [5]:

```
data['Daily Return'] = data['Close'].pct_change()
data['Strategy Return'] = data['Daily Return'] * data['Position']

# Cumulative Returns
data['Cumulative Market Return'] = (1 + data['Daily Return']).cumprod()
data['Cumulative Strategy Return'] = (1 + data['Strategy Return']).cumprod()
```

Plot the Results 📊

In [6]:

```
plt.figure(figsize=(14, 8))
plt.plot(data['Cumulative Market Return'], label='Market Return')
plt.plot(data['Cumulative Strategy Return'], label='Strategy Return')
plt.title(f"{ticker} - SMA Crossover Strategy")
plt.xlabel('Date')
plt.ylabel('Cumulative Returns')
plt.legend()
plt.grid(True)
plt.show()
```



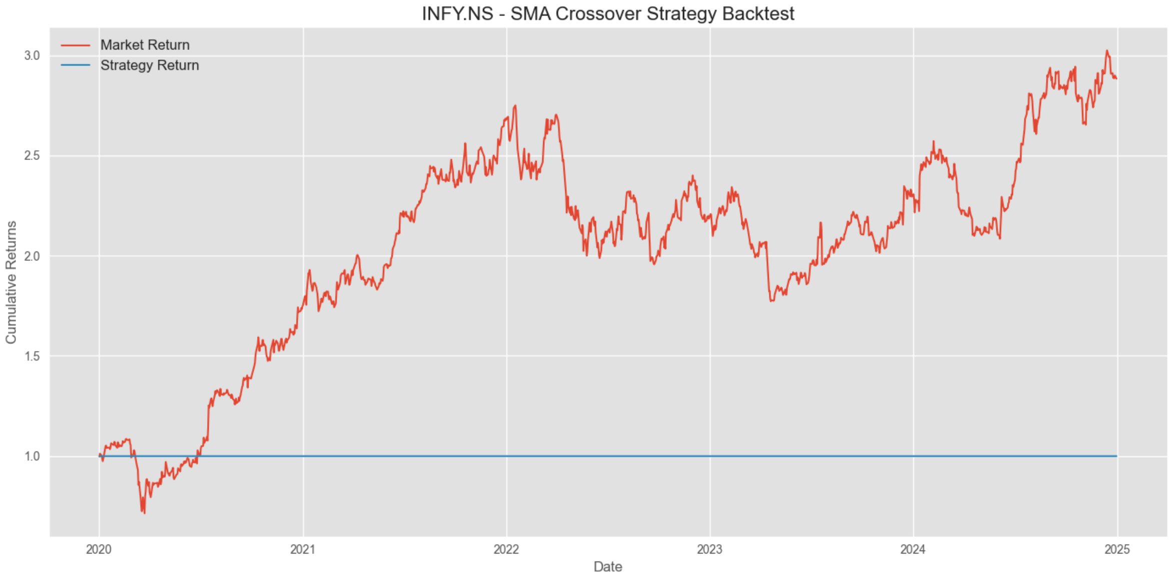
MOVING AVERAGE FOR HDFC BANK, INFOSYS TRADING STRATIGY📊

In [7]:

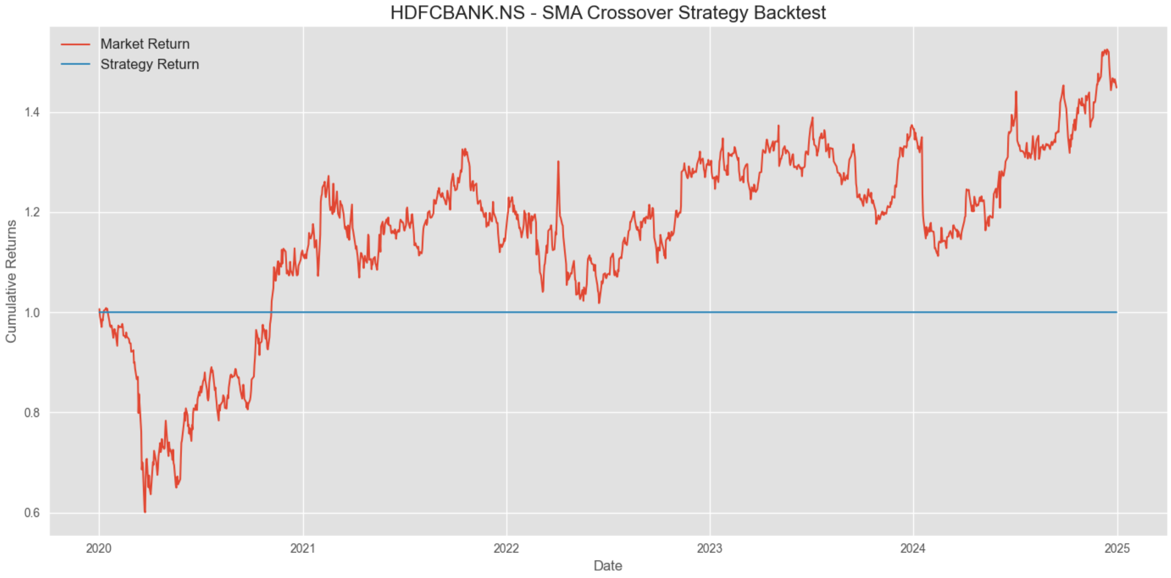
```
import yfinance as yf
import pandas as pd
import numpy as np
```

```python
import matplotlib.pyplot as plt

plt.style.use('ggplot')  # or 'fivethirtyeight' for modern look

def backtest_sma_strategy(ticker, start='2020-01-01', end='2024-12-31'):
    # Download data
    data = yf.download(ticker, start=start, end=end)

    # Calculate SMAs
    data['SMA50'] = data['Close'].rolling(window=50).mean()
    data['SMA200'] = data['Close'].rolling(window=200).mean()

    # Generate signals
    data['Signal'] = 0
    data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1,
    data['Position'] = data['Signal'].shift(1)

    # Calculate returns
    data['Daily Return'] = data['Close'].pct_change()
    data['Strategy Return'] = data['Daily Return'] * data['Position']
    data['Cumulative Market Return'] = (1 + data['Daily Return']).cumprod()
    data['Cumulative Strategy Return'] = (1 + data['Strategy Return']).cumprod()

    # Plotting
    plt.figure(figsize=(14, 7))
    plt.plot(data['Cumulative Market Return'], label='Market Return', linewidth=
    plt.plot(data['Cumulative Strategy Return'], label='Strategy Return', linewi
    plt.title(f"{ticker} - SMA Crossover Strategy Backtest", fontsize=16)
    plt.xlabel("Date", fontsize=12, )
    plt.ylabel("Cumulative Returns", fontsize=12)
    plt.legend(fontsize=12)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

    return data

# Run for Infosys and HDFC Bank
data_infy = backtest_sma_strategy("INFY.NS")      # Infosys
data_hdfc = backtest_sma_strategy("HDFCBANK.NS")  # HDFC Bank
```

```
[*********************100%***********************]  1 of 1 completed
C:\Users\Prajwal\AppData\Local\Temp\ipykernel_18240\2114212364.py:18: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1, -1)
```



```
[*********************100%***********************]  1 of 1 completed
C:\Users\Prajwal\AppData\Local\Temp\ipykernel_18240\2114212364.py:18: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1, -1)
```



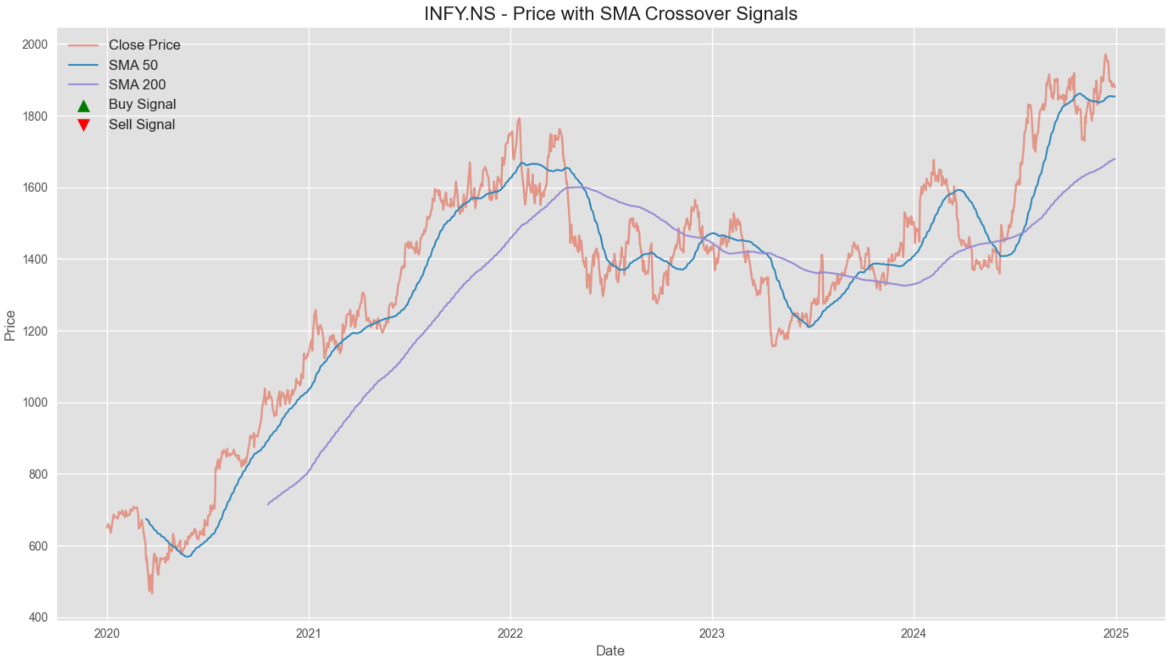
✅ Updated Function with Buy/Sell Markers

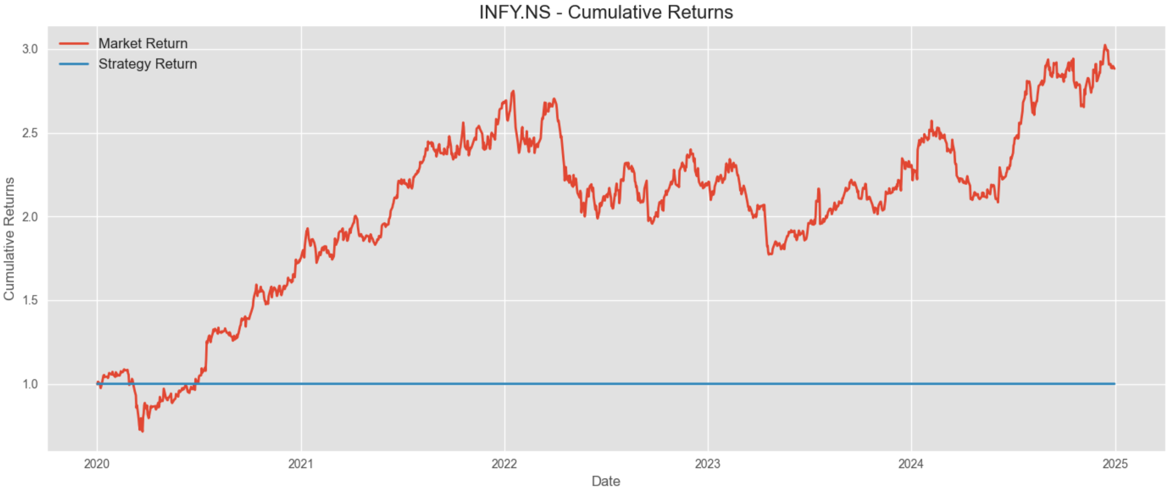
In [8]:

```python
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

plt.style.use('ggplot')

def backtest_sma_strategy(ticker, start='2020-01-01', end='2024-12-31'):
    # Download data
    data = yf.download(ticker, start=start, end=end)

    # Calculate SMAs
    data['SMA50'] = data['Close'].rolling(window=50).mean()
    data['SMA200'] = data['Close'].rolling(window=200).mean()

    # Generate signals
    data['Signal'] = 0
    data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1,
    data['Position'] = data['Signal'].shift(1)

    # Calculate returns
    data['Daily Return'] = data['Close'].pct_change()
    data['Strategy Return'] = data['Daily Return'] * data['Position']
    data['Cumulative Market Return'] = (1 + data['Daily Return']).cumprod()
    data['Cumulative Strategy Return'] = (1 + data['Strategy Return']).cumprod()

    # Identify Buy and Sell points
    buy_signals = data[(data['Position'] == 1) & (data['Position'].shift(1) == -
    sell_signals = data[(data['Position'] == -1) & (data['Position'].shift(1) ==

    # Plot price with SMAs and markers
    plt.figure(figsize=(14, 8))
    plt.plot(data['Close'], label='Close Price', alpha=0.5)
    plt.plot(data['SMA50'], label='SMA 50', linewidth=1.5)
    plt.plot(data['SMA200'], label='SMA 200', linewidth=1.5)

    plt.scatter(buy_signals.index, buy_signals['Close'], marker='^', color='gree
    plt.scatter(sell_signals.index, sell_signals['Close'], marker='v', color='re

    plt.title(f"{ticker} - Price with SMA Crossover Signals", fontsize=16)
    plt.xlabel("Date", fontsize=12)
    plt.ylabel("Price", fontsize=12)
    plt.legend(fontsize=12)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

    # Also plot cumulative returns
    plt.figure(figsize=(14, 6))
    plt.plot(data['Cumulative Market Return'], label='Market Return', linewidth=
    plt.plot(data['Cumulative Strategy Return'], label='Strategy Return', linewi
    plt.title(f"{ticker} - Cumulative Returns", fontsize=16)
    plt.xlabel("Date", fontsize=12)
    plt.ylabel("Cumulative Returns", fontsize=12)
    plt.legend(fontsize=12)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

    return data
```

In [9]:
```
backtest_sma_strategy("INFY.NS")
backtest_sma_strategy("HDFCBANK.NS")
```

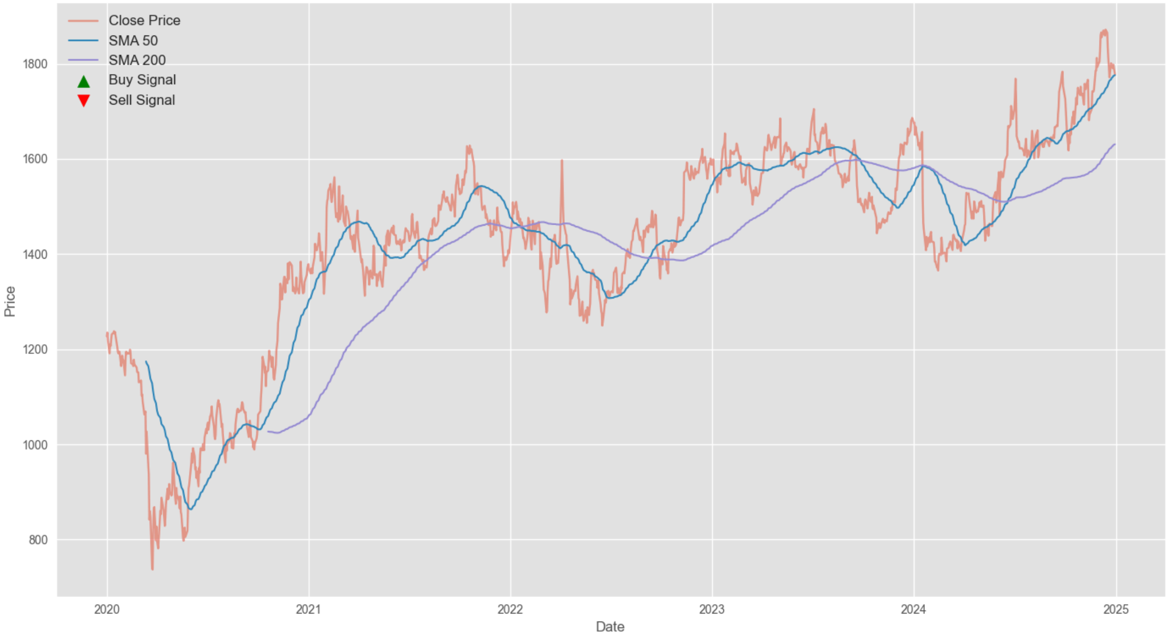
```
[*********************100%***********************]  1 of 1 completed
C:\Users\Prajwal\AppData\Local\Temp\ipykernel_18240\2836337225.py:18: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1, -1)
```



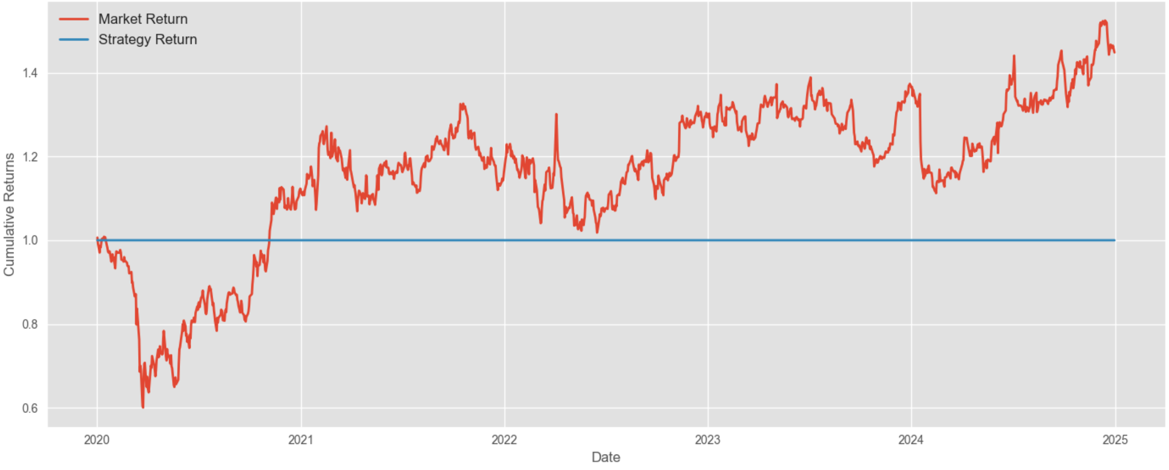


```
[*********************100%***********************]  1 of 1 completed
C:\Users\Prajwal\AppData\Local\Temp\ipykernel_18240\2836337225.py:18: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['Signal'][50:] = np.where(data['SMA50'][50:] > data['SMA200'][50:], 1, -1)
```

HDFCBANK.NS - Price with SMA Crossover Signals



HDFCBANK.NS - Cumulative Returns



Out[9]:

Price	Close	High	Low	Open	Volume
Ticker	HDFCBANK.NS	HDFCBANK.NS	HDFCBANK.NS	HDFCBANK.NS	HDFCBANK.NS
Date					
2020-01-01	1227.428711	1228.772705	1219.748882	1225.028764	1836849
2020-01-02	1235.252441	1236.452415	1227.812607	1227.812607	3068583
2020-01-03	1217.636963	1233.572585	1213.029018	1230.884598	5427775
2020-01-06	1191.285400	1211.301048	1186.533553	1209.573040	5445093
2020-01-07	1210.149048	1220.564793	1202.133250	1208.517131	7362247
...	...	...	...	...	...
2024-12-23	1801.000000	1806.000000	1781.300049	1782.000000	5522296
2024-12-24	1798.099976	1808.699951	1789.550049	1802.050049	7242917
2024-12-26	1790.750000	1812.000000	1780.750000	1798.550049	5240839
2024-12-27	1798.250000	1805.949951	1789.099976	1794.800049	3629665
2024-12-30	1777.900024	1815.000000	1771.000000	1792.199951	11111109

1237 rows × 13 columns

In [ ]:

In [ ]: