

IN THIS CODE I MADE A SYSTEM THAT COLLECTS DATA FROM MONEY CONTROL API AND THEN CONVERTS DATA INTO DATAFRAME

HERE WE CAN PREDICT SENSEX MOMENT AND PLAN A TRADE IN REAL TIME MARKET

IN THE NOTEBOOK IS THE REPRESENTATION OF HOW WE ANALYSE THE MARKET BY REALTIME DATA

IN THIS CODE WE MAKE USE OF ALGO TRADING AND IDENTIFY POTNETIAL TRADE BY MY TWO FAVORITE TECHNIQUES

1) MA CROSS

THE DATA IS COLLECTED USING MONEY CONTROL API AND THEN CONVERTED INTO DATAFRAME. HERE THE DATA OF SENSEX IS IN 5 MIN FORMAT

```
In [51]: import pandas as pd

import matplotlib.pyplot as plt
data_str = "70828.57,70833.9,70793.79,70884.08,70863.77,70901.9,70951.12,70865.

data_list = data_str.split(',')

df = pd.DataFrame(data_list, columns=['Value'], dtype=float)
```

FOR THE CODE WE BELOW WE CALCULATE IF 9 MA CROSSES 21 MA IN UP DIRECTION IT

WILL CREATE A BUY ALERT AND SEND EMAIL TO OUR MAIL ID

```
In [92]: import pandas as pd
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
import matplotlib.pyplot as plt

# Function to send an email
def send_email(subject, body):
    sender_email = "iamgr8@gmail.com"
    sender_password = "213456789"
    receiver_email = "iamgr8@gmail.com"

    message = MIMEMultipart()
    message["From"] = sender_email
    message["To"] = receiver_email
    message["Subject"] = subject

    message.attach(MIMEText(body, "plain"))

    with smtplib.SMTP("smtp.gmail.com", 587) as server:
        server.starttls()
        server.login(sender_email, sender_password)
        server.sendmail(sender_email, receiver_email, message.as_string())

df['Date'] = range(len(df))

df['9_MA'] = df['Value'].rolling(window=9).mean()
df['21_MA'] = df['Value'].rolling(window=21).mean()

df['CrossAbove'] = (df['9_MA'] > df['21_MA']) & (df['9_MA'].shift(1) <= df['21_

df.dropna(subset=['CrossAbove'], inplace=True)

for index, row in df.iterrows():
    if row['CrossAbove']:
        buy_price = row['Value']
        buy_date = row['Date']
        print(f"Buy Alert: Date - {buy_date}, Price - {buy_price}")
        send_email(f"Buy Alert at Date {buy_date}", f"Buy Alert: Date - {buy_da

plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df['Value'], label='Value', linewidth=2)

plt.scatter(
    df[df['CrossAbove']]['Date'],
    df[df['CrossAbove']]['Value'],
    marker='^',
```

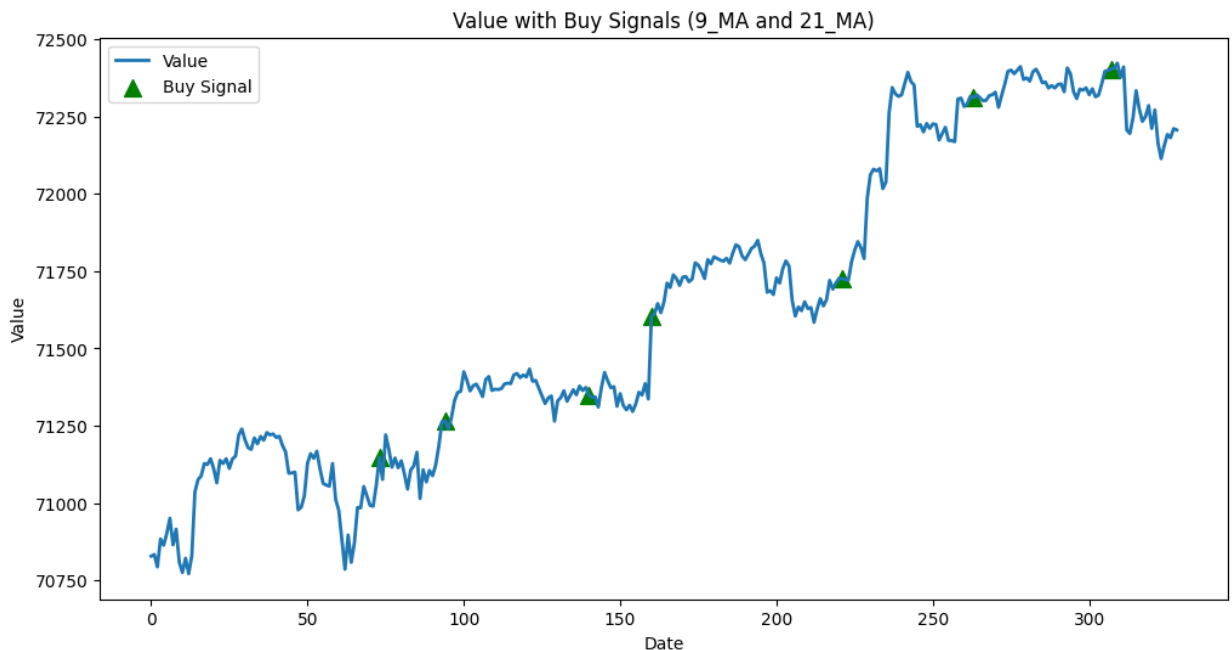
```

        color='g',
        label='Buy Signal',
        s=100
    )

plt.title('Value with Buy Signals (9_MA and 21_MA)')
plt.xlabel('Date')
plt.ylabel('Value')
plt.legend()
plt.show()

```

Buy Alert: Date - 73, Price - 71147.61
 Buy Alert: Date - 94, Price - 71266.6
 Buy Alert: Date - 140, Price - 71348.9
 Buy Alert: Date - 160, Price - 71604.12
 Buy Alert: Date - 221, Price - 71725.92
 Buy Alert: Date - 263, Price - 72311.3
 Buy Alert: Date - 307, Price - 72403.89



IN THIS CODE WE CALCULATE PERCENTAGE CHANGE FROM BUY TO SELL AND RESULTS ARE GOOD

```

In [53]: df['Date'] = range(len(df))

df['9_MA'] = df['Value'].rolling(window=9).mean()
df['21_MA'] = df['Value'].rolling(window=21).mean()

df['Buy_Signal'] = (df['9_MA'] > df['21_MA']) & (df['9_MA'].shift(1) <= df['21_
df['Sell_Signal'] = (df['9_MA'] < df['21_MA']) & (df['9_MA'].shift(1) >= df['21_

buy_signals = df[df['Buy_Signal']]
sell_signals = df[df['Sell_Signal']]

```

```

buy_price = None
sell_price = None

for index, row in df.iterrows():
    if row['Buy_Signal']:
        buy_price = row['Value']
        print(f"Buy Alert: Date - {row['Date']}, Price - {buy_price}")
    elif row['Sell_Signal'] and buy_price is not None:
        sell_price = row['Value']
        percentage_change = ((sell_price - buy_price) / buy_price) * 100
        print(f"Sell Alert: Date - {row['Date']}, Price - {sell_price}, Percent

        buy_price = None

    if percentage_change < 0:
        print("Negative Percentage Change Alert!")

plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df['Value'], label='Value', linewidth=2)

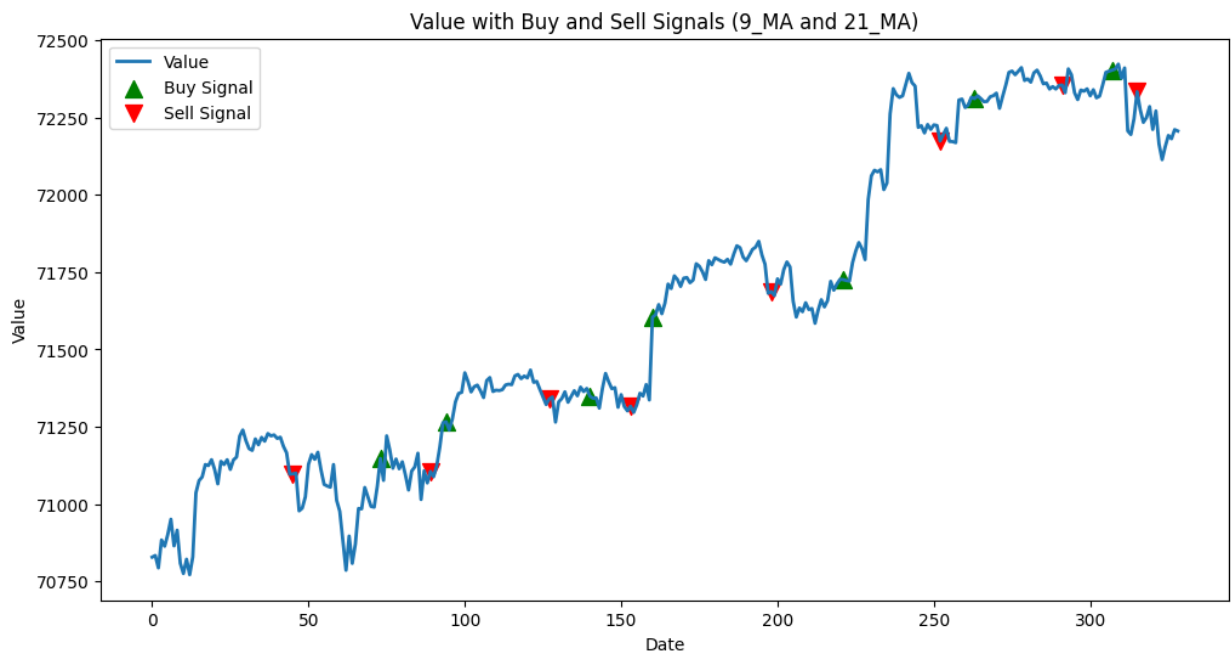
# Plot Buy signals
plt.scatter(
    buy_signals['Date'],
    buy_signals['Value'],
    marker='^',
    color='g',
    label='Buy Signal',
    s=100 # Marker size
)

# Plot Sell signals
plt.scatter(
    sell_signals['Date'],
    sell_signals['Value'],
    marker='v',
    color='r',
    label='Sell Signal',
    s=100 # Marker size
)

plt.title('Value with Buy and Sell Signals (9_MA and 21_MA)')
plt.xlabel('Date')
plt.ylabel('Value')
plt.legend()
plt.show()

```

Buy Alert: Date - 73, Price - 71147.61
 Sell Alert: Date - 89, Price - 71105.69, Percentage Change: -0.06%
 Negative Percentage Change Alert!
 Buy Alert: Date - 94, Price - 71266.6
 Sell Alert: Date - 127, Price - 71340.08, Percentage Change: 0.10%
 Buy Alert: Date - 140, Price - 71348.9
 Sell Alert: Date - 153, Price - 71316.64, Percentage Change: -0.05%
 Negative Percentage Change Alert!
 Buy Alert: Date - 160, Price - 71604.12
 Sell Alert: Date - 198, Price - 71687.11, Percentage Change: 0.12%
 Buy Alert: Date - 221, Price - 71725.92
 Sell Alert: Date - 252, Price - 72174.19, Percentage Change: 0.62%
 Buy Alert: Date - 263, Price - 72311.3
 Sell Alert: Date - 291, Price - 72355.47, Percentage Change: 0.06%
 Buy Alert: Date - 307, Price - 72403.89
 Sell Alert: Date - 315, Price - 72333.72, Percentage Change: -0.10%
 Negative Percentage Change Alert!



IN THIS CODE IF PRICE CROSSES RESISTANCE
ALERT IS CREATED

```

In [54]: resistance_levels = [71000, 71500, 72000]

for level in resistance_levels:
    df[f'AboveResistance_{level}'] = df['Value'] > level

resistance_cross_alerts = pd.DataFrame()
for level in resistance_levels:
    resistance_cross_alerts = pd.concat([resistance_cross_alerts, df[df[f'AboveResistance_{level}'] == True]])

plt.figure(figsize=(12, 6))
plt.plot(df['Value'], label='Close Price', linewidth=2)

for level in resistance_levels:
    cross_above_resistance = resistance_cross_alerts[resistance_cross_alerts[f'AboveResistance_{level}'] == True]
    plt.scatter(cross_above_resistance, df.loc[cross_above_resistance, 'Value'])
  
```

```

plt.xlabel('Index')
plt.ylabel('Price')
plt.title('Closing Prices with Cross Above Resistance Alerts')
plt.legend()
plt.grid(True)

plt.show()

print("\nCross Above Resistance Alerts:")
for level in resistance_levels:
    print(resistance_cross_alerts[resistance_cross_alerts[f'AboveResistance_{level}']])

```



```

Cross Above Resistance Alerts:
      Value  AboveResistance_71000
14  71036.66                    True
49  71023.57                    True
68  71053.85                    True
72  71057.48                    True
160 71604.12                    True
230 72061.37                    True
      Value  AboveResistance_71500
160 71604.12                    True
230 72061.37                    True
      Value  AboveResistance_72000
230 72061.37                    True

```

**IN THIS CODE IF PRICE CROSSES SUPPORT
ALERT IS CREATED**

```

In [55]: threshold_level = 71500

df['BelowThreshold'] = df['Value'] < threshold_level

cross_below_threshold_alerts = df[df['BelowThreshold'].diff() == 1]

plt.figure(figsize=(12, 6))

```

```

plt.plot(df['Value'], label='Close Price', linewidth=2)

cross_below_threshold = cross_below_threshold_alerts.index
plt.scatter(cross_below_threshold, df.loc[cross_below_threshold, 'Value'], mark

plt.xlabel('Index')
plt.ylabel('Price')
plt.title('Closing Prices with Cross Below Threshold Alerts')
plt.axhline(y=threshold_level, color='gray', linestyle='--', label='Threshold L
plt.legend()
plt.grid(True)

plt.show()

print("\nCross Below Threshold Alerts:")
print(cross_below_threshold_alerts[['Value', 'BelowThreshold']])

```



```

Cross Below Threshold Alerts:
      Value  BelowThreshold
160  71604.12             False

```

NOW LETS USE RSI FOR SOME TRADING

```

In [56]: import pandas as pd
import matplotlib.pyplot as plt
import ta

df['RSI'] = ta.momentum.RSIIndicator(df['Value']).rsi()

rsi_threshold = 40

rsi_cross_alerts = df[(df['RSI'] > rsi_threshold) & (df['RSI'].shift(1) <= rsi_

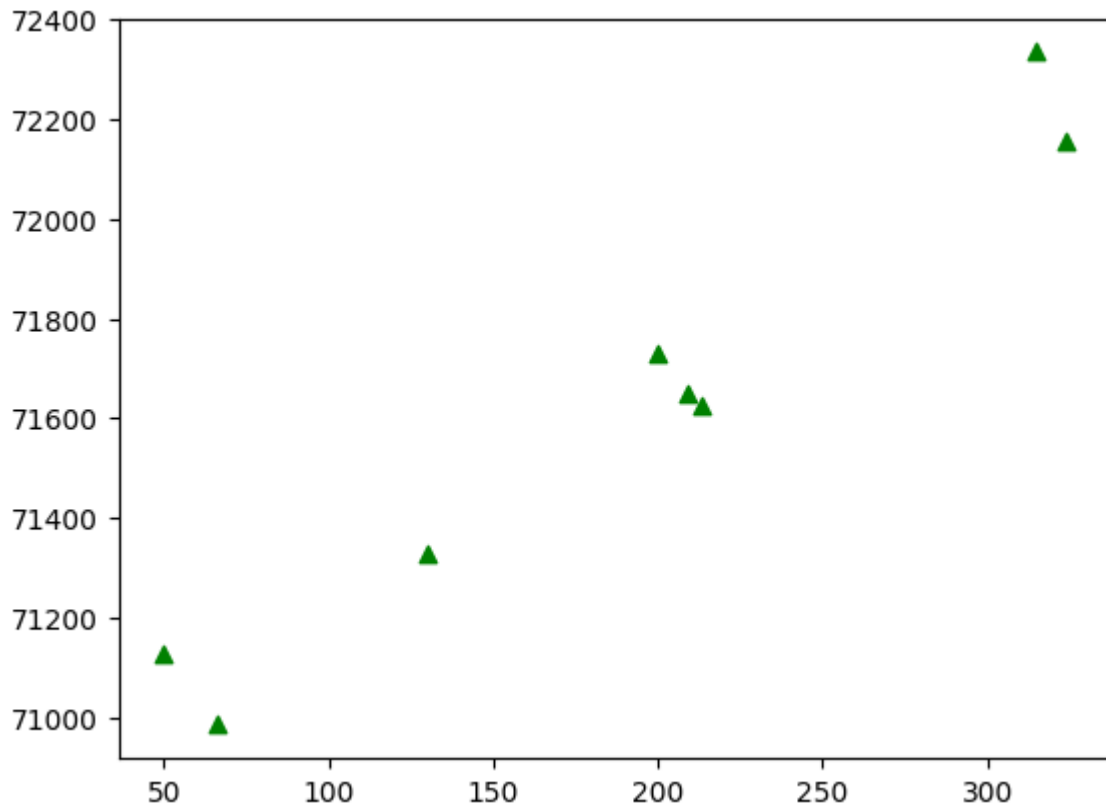
rsi_cross_above_threshold = rsi_cross_alerts.index
plt.scatter(rsi_cross_above_threshold, df.loc[rsi_cross_above_threshold, 'Value

```

```
print("\nRSI Cross Above Threshold Alerts:")
print(rsi_cross_alerts[['Value', 'RSI']])
```

RSI Cross Above Threshold Alerts:

	Value	RSI
50	71128.80	51.921951
66	70986.03	48.095461
130	71330.54	47.703187
200	71728.52	47.938613
209	71651.26	42.767023
213	71627.58	42.681592
315	72333.72	49.839116
324	72156.18	40.241468



HERE IN THIS PROJECT I MAKE USE OF
MACHING LEARNING WITH TRADING TO
CREATE A ALGO TRADING IIKE SYSTEM

THIS CODE CAN BE IMPROVES AND MORE
ALGORITHMS OR TECHNIQUES CAN BE
ADDED TO IT