Problem Statement :

Objective

The primary goal of this project is to enhance an existing trading strategy by developing a more robust machine learning algorithm

Deliverables

- Develop a predictive algorithm that can classify the trend
- Ensure minimal accuracy disparity between training and test datasets

Details:
- Trend_initial: Represents the initial trend, capturing both upward and downward movements
- Trend_final: The dependent variable, derived from Trend_initial, containing only points where the overall trend is upward
- Independent Variables: The other variables, along with Trend_initial, serve as the independent variables for the model

- Goal: To eliminate data points where Trend_initial is 1 and Trend_final is 0, as these points indicate a downward overall trend
- Model building: Build a machine learning model that can accurately classify the dependent variable, Trend_final, based on the independent variables

Data

X Input_data.xlsx

	A	B	C	D	E	F	G	H	I	J
1	Date-Time	Open	High	Low	Close	cci_50	Close_supertrend_diff	ATR_EMA_diff	Trend_initial	Trend_final
2	04/01/11 9:55	6150.1	6150.1	6136	6136.3	-283.26	-0.01	2.53	1	0
3	04/01/11 10:00	6135.9	6145.75	6134.85	6145.4	-240.60	0.00	2.57	1	0
4	04/01/11 10:05	6145.55	6150	6145.2	6148.25	-161.85	0.00	2.24	1	0
5	04/01/11 10:10	6148.4	6151.4	6145.5	6151.3	-137.16	0.00	2.11	1	0
6	04/01/11 10:15	6151.55	6151.85	6149.15	6149.95	-122.81	0.00	1.60	1	0
7	04/01/11 10:20	6149.9	6154	6149.9	6153.4	-98.37	0.00	1.19	1	0
8	04/01/11 10:25	6153.6	6154.55	6151.75	6151.8	-91.42	0.00	0.68	1	0
9	04/01/11 10:30	6151.7	6151.85	6142.5	6143.15	-146.20	0.00	-0.49	1	0
10	04/01/11 10:35	6143.35	6144.1	6140.8	6140.95	-168.13	0.00	-0.70	1	0
11	04/01/11 10:40	6141.1	6141.65	6135.9	6140.95	-174.69	0.00	-0.96	1	0

Key Metrics:

CCI (Commodity Channel Index):

CCI helps traders identify when a market is overbought or oversold.
It measures the current price relative to a moving average over a given period of time.
A high CCI indicates overbought conditions, suggesting a potential downward trend, while a low CCI indicates oversold conditions, suggesting a potential upward trend.

- Formula: $CCI = \frac{(Typical\ Price - SMA)}{(0.015 \times Mean\ Deviation)}$

- Where:

    - Typical Price = (High + Low + Close) / 3

    - SMA = Simple Moving Average of Typical Price over a chosen period (usually 20 periods)

    - Mean Deviation = Average of the absolute difference between each Typical Price and the SMA over the chosen period

The constant 0.015 is used for scaling purposes to ensure that 70–80% of CCI values fall between -100 and +100

RSI (Relative Strength Index):

RSI measures the speed and change of price movements.
It ranges from 0 to 100.
Traditionally, when RSI is above 70, it indicates overbought conditions, suggesting a potential downward trend, and when it's below 30, it indicates oversold conditions, suggesting a potential upward trend.

- Formula: $RSI = 100 - \frac{100}{1 + RS}$

- Where:

    - RS = Average of x days' up closes / Average of x days' down closes

    - Typically, x is 14 days

To Calculate the Average Gain (AG) and Average Loss (AL) over a specified period:
AG = Average gain over the specified period (usually 14 days)
AL = Average loss over the specified period (usually 14 days)

To calculate AG and AL:
For each day in the specified period, calculate the price change:
If the price increased that day, the gain is the difference between the current day's close price and the previous day's close price.
If the price decreased that day, the loss is the difference between the previous day's close price and the current day's close price.

Calculate the average gain (AG) and average loss (AL) over the specified period:
AG = (Sum of gains over the specified period) / (Number of days in the specified period)
AL = (Sum of losses over the specified period) / (Number of days in the specified period)

Calculate the Relative Strength (RS):

RS = AG / (AL + 0.0001)
Small constant 0.0001 is added to avoid division by zero error, in case of AL = 0

Calculate the RSI:
RSI = 100 - (100 / (1 + RS))

Identify the Gain for Each Period: For each period, calculate the gain, which represents the price increase from the previous period. The gain is calculated as the difference between the current closing price and the previous closing price if the current price is higher.

Gain=Current close price−Previous close price

If the current close price is lower than the previous close price, the gain for that period is zero.

ATR (Average True Range):

ATR measures market volatility.
It calculates the average range between the high and low prices over a given period.
Higher ATR values indicate higher volatility, while lower values suggest lower volatility.

- Formula: $ATR = \frac{1}{n} \sum_{i=1}^{n} TR_i$

- Where:

  - TR_i = Max(High_i - Low_i, |High_i - Close_{i-1}|, |Low_i - Close_{i-1}|)

  - Typically, n is 14 periods

Supertrend:

Supertrend is a trend following indicator.
It plots a line either above or below the price, indicating the current trend direction.
When the Supertrend line is below the price, it suggests a bullish (upward) trend, and when it's above the price, it suggests a bearish (downward) trend.

- Formula (for Uptrend): $Basic\ Upper\ Band = (High + Low)/2 + Multiplier \times ATR$

- Formula (for Downtrend): $Basic\ Lower\ Band = (High + Low)/2 - Multiplier \times ATR$

- Where:

  - ATR = Average True Range over a chosen period

  - Multiplier = typically 2 or 3

EMA (Exponential Moving Average):

EMA is a type of moving average that gives more weight to recent prices.
It smoothes out price data to identify trends more easily.
Shorter EMA periods respond more quickly to price changes, while longer periods offer smoother trends.

- Formula: $EMA = (Close - EMA_{prev}) \times \frac{2}{n+1} + EMA_{prev}$

- Where:

  - Close = Closing price of the current period

  - EMA_{prev} = EMA of the previous period

  - n = number of periods

For the 1st ever value of EMA prev calculation, we consider the closing value as EMA prev directly.

MACD (Moving Average Convergence Divergence):

MACD is a trend-following momentum indicator.
It consists of two lines: the MACD line and the signal line.
MACD is created by subtracting **12 day EMA value** from **26 day EMA value**
Signal is created using **9 day EMA value**
When the MACD line crosses above the signal line, it indicates a bullish trend, and when it crosses below the signal line, it indicates a bearish trend.

- Formula (MACD Line): $MACDLine = 12 - dayEMA - 26 - dayEMA$

- Formula (Signal Line): $SignalLine = 9 - dayEMA$

Here's an example calculation:

1. **Gather Data**: *Let's consider the following closing prices for a stock over 12 days:*
   - *Day 1: $10.00 Day 2: $10.50 Day 3: $11.00 Day 4: $11.50*
   - *Day 5: $12.00 Day 6: $12.50 Day 7: $13.00 Day 8: $13.50*
   - *Day 9: $14.00 Day 10: $13.50 Day 11: $13.00 Day 12: $12.50*
2. **Calculate the 12-day EMA (Exponential Moving Average)**:
   - *We'll use the formula:*
     - *EMA=(Close − EMA prev)× 2/(n+1) + EMAprev*
     - *Close = Closing price of the current day*
     - *EMA prev = EMA value from the previous day*
     - *n = number of periods (in this case, 12 days)*
3. *Let's calculate the 12-day EMA for each day:*
   - *For Day 1, since it's the first data point, the EMA is equal to the closing price: EMA12=10.00*
   - *For Day 2, using the formula: EMA12=(10.50−10.00)×212+1+10.00=10.07*
   - *Continue this process for each subsequent day until you have the 12-day EMA for all 12 days.*
4. **Calculate the 26-day EMA**:
   - *Follow the same process as above, but this time, use a 26-day period.*
5. **Calculate the MACD Line**:
   - *Subtract the 26-day EMA from the 12-day EMA to get the MACD line value for each day.*
   - *For example, for Day 12: MACD=EMA12−EMA26*
6. **Calculate the 9-day EMA of the MACD Line (Signal Line)**:
   - *Calculate the EMA of the MACD line using a 9-day period.*
7. **Plot the MACD Line and Signal Line**:
   - *Plot both lines on a chart to visualise the MACD indicator.*

OBV (On-Balance Volume):

OBV measures buying and selling pressure.
It adds volume on up days and subtracts volume on down days.
Rising OBV suggests accumulation (buying pressure), while falling OBV suggests distribution (selling pressure).

- Formula:

  - If Close > Close_{prev}: OBV_{today} = OBV_{prev} + Volume_{today}

  - If Close < Close_{prev}: OBV_{today} = OBV_{prev} – Volume_{today}

  - If Close = Close_{prev}: OBV_{today} = OBV_{prev}

  - Where:

    - Close = Closing price of the current period

    - Close_{prev} = Closing price of the previous period

    - Volume_{today} = Volume of the current period

In On-Balance Volume (OBV), volume refers to the trading volume of a financial asset (such as a stock, commodity, or currency pair) over a given period.

Volume represents the total number of shares or contracts traded during a specific timeframe, such as a day, week, or month. It's an important indicator in technical analysis because it provides insight into the strength or weakness of a price trend.

In OBV, volume is used to calculate the cumulative total of positive and negative volume changes. When the closing price of the asset is higher than the previous day's closing price, the volume for that day is considered positive. Conversely, when the closing price is lower than the previous day's closing price, the volume is considered negative.

By adding the positive volume on up days and subtracting the negative volume on down days, OBV creates a cumulative total that reflects buying and selling pressure. Rising OBV suggests accumulation (buying pressure), while falling OBV suggests distribution (selling pressure).

So, in summary, volume in OBV refers to the total volume of trades over a specified period, and it's used to determine the flow of funds into or out of an asset.

Calculating the Exponential Moving Average (EMA) of the Average True Range (ATR)

This involves applying the EMA formula to the series of ATR values over a specified period. Here's how you can calculate the EMA of the ATR:

1. Calculate the ATR for each period: Use the steps mentioned earlier to calculate the ATR for each period.
2. Choose a smoothing factor (alpha): The smoothing factor, often denoted as alpha (α), determines the weight given to recent ATR values in the EMA calculation. A common smoothing factor used for EMA calculations is 2 / (n + 1), where n is the number of periods. For example, if you are using a 14-period ATR, the smoothing factor would be 2 / (14 + 1) = 0.1333.
3. Calculate the initial EMA of ATR: To start the EMA calculation, you'll need an initial EMA value. This can be the simple moving average (SMA) of the first n ATR values. For example, if you're using a 14-period ATR, calculate the SMA of the first 14 ATR values.
4. Calculate subsequent EMA values: Once you have the initial EMA value, you can calculate the EMA of the ATR for subsequent periods using the following formula:
   - $EMA_{today} = (ATR\ today - EMA\ yesterday) \times \alpha + EMA\ yesterday$
   - ATR today = ATR value for the current period
   - EMA yesterday = EMA value from the previous period
   - α = Smoothing factor (calculated in step 2)
5. Repeat step 4 for each subsequent period: Continue this process for each period, using the previous EMA value and the current ATR value to calculate the new EMA value.

Approach

## 1. Data cleaning

Total records in data were 2,46,759.
As per the requirement, records with trend_initial=1 and trend_final=0 were removed, reducing the records to 2,10,475.
For each day, data has been captured starting 9:20 every 5 mins, till 15:30, except for a few days (eg. 2015-12-21, 2020-3-13, 2021-2-24, 2021-12-6, 2023-11-6).
None of such days have been imputed assuming the entire data to be continuous.

2. Exploratory data analysis

- Open, Close, High, Low, ATR, SMA50, EMA50 are high correlated
- Trend initial and Trend Final are highly correlated
- Trend initial and Trend Final both have med to high negative correlated to CCI50
- Close, CCI50, TR, Close supertrend diff, ATR_EMA_diff can be used as features for the model
- Trend Final can be used as Target variable
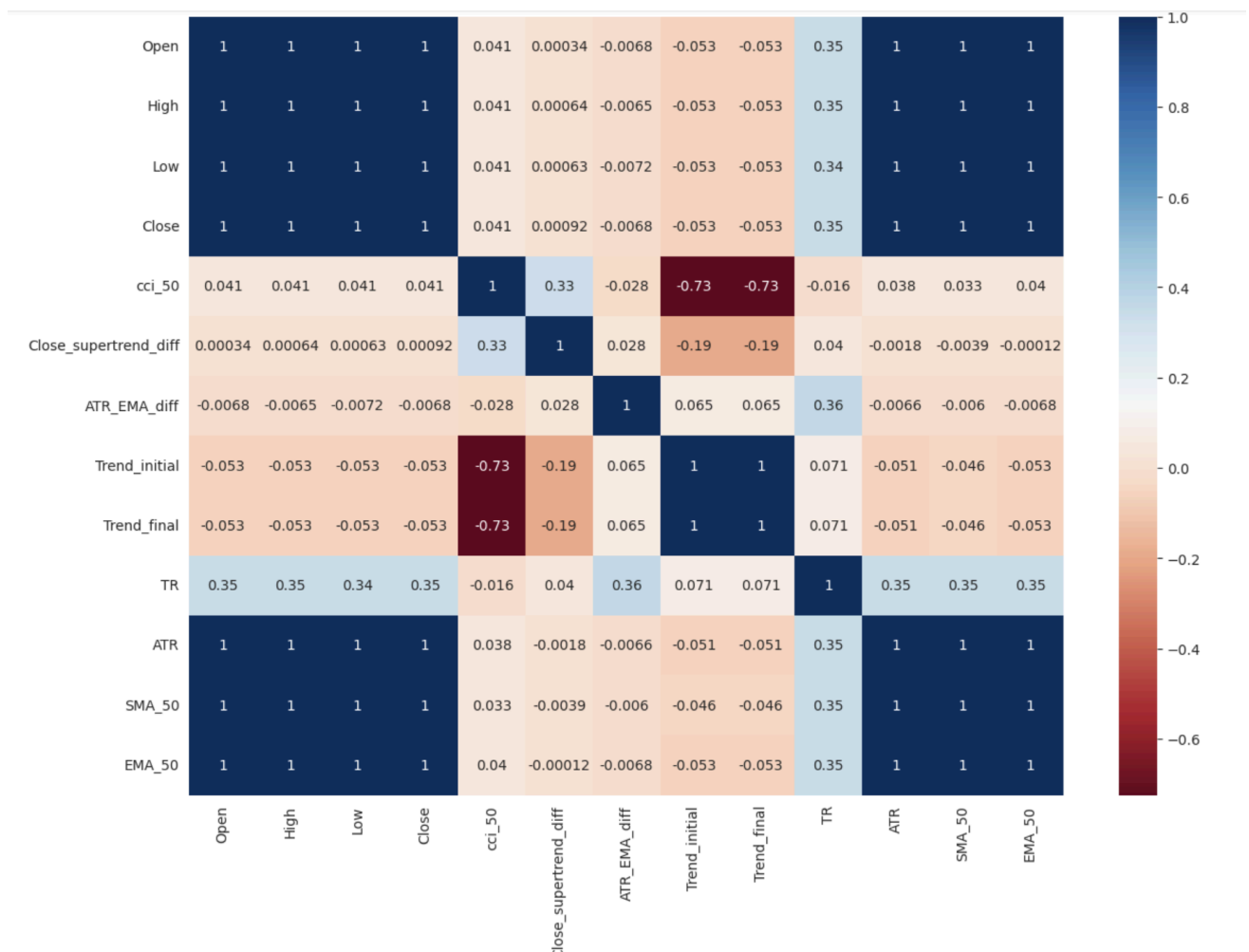
3. Creation of additional metrics

- TR has been created, also used as a feature
- Date time like *day of week, day of month, quarter, etc*. fields were also created, but these didn't improve the model prediction, hence were removed as a feature in the final model

4. Model implementation

LSTM (Long short term memory model) was used for this problem, as it is type of recurrent neural network with a memory cell to use output of previous time step as an input to next time step.

10 LSTM layers and 1 dense layer has been added with sigmoid activation function for both layers. And, binary cross entropy with Adam optimiser has been used as loss function.

80:20 split was used for data splitting into train and test data.

∨ LSTM model development

```python
[261] def LSTM_model_training(X_train, y_train, _nepoch, _nbatchsize, _verbose):

        """
        ---------------------------------------------------
        Building the LSTM Model
        ---------------------------------------------------
        """

        lstm = Sequential()
        lstm.add(LSTM(10, input_shape=(1, X_train.shape[2]), activation='sigmoid', return_sequences=False))
        lstm.add(Dense(1, activation='sigmoid'))
        lstm.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
        print(lstm.summary())

        timestamp = str(datetime.datetime.now())[:13].replace(' ','_H')
        modelfilename = 'model/cnn_lstm_rating_'+timestamp+'.h5'
        modelfilename = os.path.join(cwdir,DrivePath,modelfilename)
        print("Filepath :", modelfilename)

        callbacks_lstm = [callbacks.EarlyStopping(monitor='acc', patience=3),
                          callbacks.ModelCheckpoint(filepath=modelfilename, monitor='val_loss', save_best_only=True)
                          ]

        history = lstm.fit(X_train, y_train, epochs= _nepoch, batch_size= _nbatchsize, verbose=_verbose, shuffle=False, callbacks=callbacks_lstm)

        return lstm
```

```
Model: "sequential_10"
_____
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 10)                640

 dense_8 (Dense)             (None, 1)                 11

=================================================================
Total params: 651 (2.54 KB)
Trainable params: 651 (2.54 KB)
Non-trainable params: 0 (0.00 Byte)
```
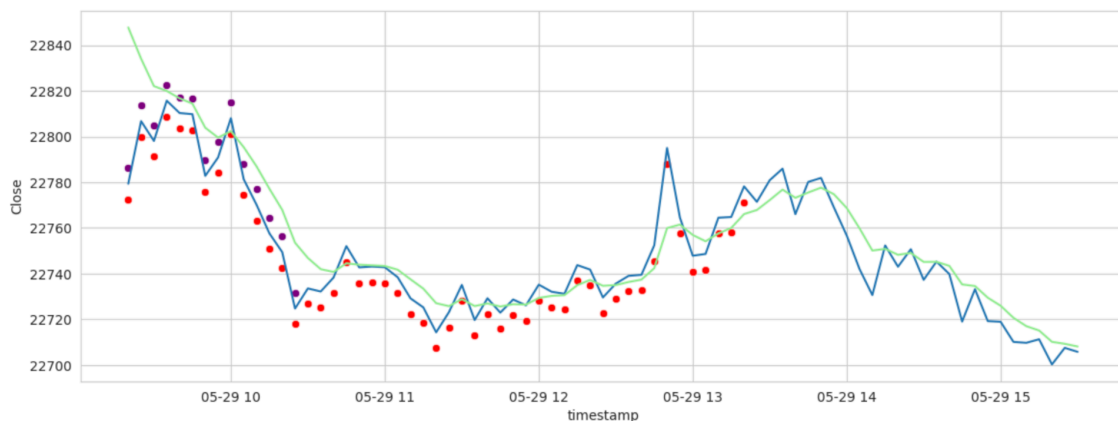
5. Key Takeaway

Precision of the model for predicting upward trend was at 85% with a recall of 80%. The weighted average F1-score was at 89%.

```
               precision    recall  f1-score   support

           0       0.90      0.93      0.92     23337
           1       0.85      0.80      0.82     11742

    accuracy                           0.89     35079
   macro avg       0.88      0.86      0.87     35079
weighted avg       0.89      0.89      0.89     35079
```

The below plot shows trend curve for a specific day - '2024-05-29'
- Close
- EMA50
- upward trend positions (actual and predicted)

This can be used as a curve in production to recommend upward trend positions.

6. Code Repo

- https://github.com/PankajShukla/Stock-Market-Trend-Prediction.git

The document ends here.