classification model → predict stock price
↑ or ↓ or range of technical indicators.

Decision Trees:
fundamental building blocks of Random Forest.

a flowlike chart structure.
each node ← test a particular attribute of an object.

e.g.: A person → man? woman? (decision node) → male (leaf node)
→ female (leaf node)
or (terminal node)

1st decision node: Root Node.

Root Node: entire population. starting point.
Splitting: dividing a node into 2 or more sub nodes.
e.g.: gender.

Pruning: remove subnodes of a decision node, ... pruning.

Branch / Sub-tree: A sub-section of entire tree is called branch or sub-tree.

Parent and Child Node: A node, which is divided into sub Nodes is called parent Node (Child Node ~)

An Ensemble learning model is a model in which decisions are used from multiple models to improve the overall performance of the model.

e.g.

Bagging in rndm frst mdl.

Random Forest: ← we need it Because: weaknesses of decision node

weakness of decision_tree:

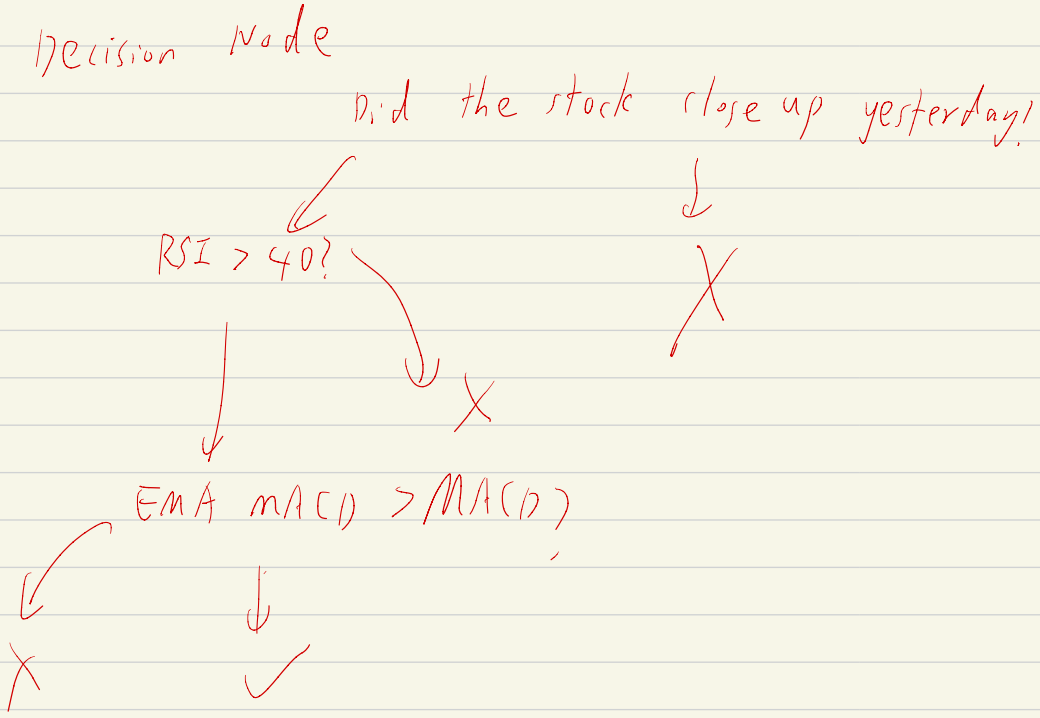
1. Instability: every small changes → (can have dramatic change) overall structure.
2. Inaccurate.
3. more levels > decision tree
   categorical variable
4. calculations → complex.

Supervised Learning:

(1) unsupervised learning: don't supervise the model (... instead ...) allow it to discover information on its own.
(unlabelled)

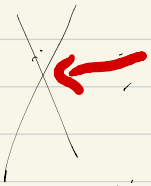
(2) with supervised learning, we provide the model with a "LABELED" dataset which tells the model what the "correct" value it should be.

e.g. Random forest



bootstrap aggregation: OR bagging:

① reducing the variance of an estimated prediction function.
↳ (high variance. low-bias procedures/trees



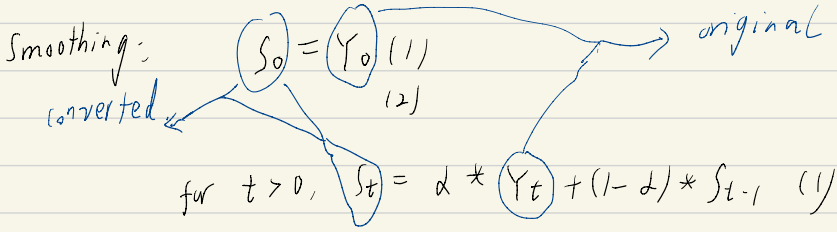
weaknesses of bagging

1. It's using bagging algorithm
2. decision Trees uses Gini-Index, a greedy algorithm to find the best result.
3. End up with trees — structurally similarly to each other.
i.e. highly-correlated.

Data preprocessing:

several ways ① API library

② directly find them in the website.



If $\alpha = 1$ The smoothed statistic becomes equal to the actual observation.

The goal of smoothing: remove the randomness and noise from our data. (Not spiky up and down graph but, instead, a smoother one)

Indicator Calculation:

Relative Strength Index (RSI)

over 70 overbought

$$RSI = \frac{100}{1 + RS}$$

To get RS (relative strength).

change in price > 0 $df < 0$ make it 0

.... < 0 ... > 0 0

Then, Relative strength $= \frac{\sum \text{down\_change}}{\sum \text{up\_change}}$.

Indication calculation: Stochastic Oscillator:

It measures the level of the closing price relative to the low-high range over a period of time.

$$K = 100 * \frac{(C - L14)}{(H14 - L14)} \quad (1)$$

C = current closing price

L14 = Lowest Low over the past 14 days

H14 = Highest High over the past 14 days.

William's R

William% R ranges from -100 to 0. When its value is above -20, it indicates a sell signal and when its value is below -80, it indicates a buy signal.

$$R = \frac{(H14 - C)}{(H14 - L14)} - 100$$

C: current closing price L14: Lowest low over the past 14 days

H14: Highest high over the past 14 days

Indicator Calculation: Moving Average Convergence Divergnece (MACD)

Definition From Paper:

EMA stands for Exponential Moving Average. When the MACD goes below the SingalLine, it indicates a sell signal. When it goes above the SignalLine, it indicates a buy signal.

Formula:

$$MACD = EMA_{12}(C) - EMA_{26}(C) \quad (1)$$

$$SignalLine = EMA_9(MACD) \quad (1)$$

$where, \quad (1)$

$MACD = \text{Moving Average Convergence Divergence} \quad (1)$

$C = \text{Closing Price} \quad (1)$

$EMA_n = \text{n day Exponential Moving Average} \quad (1)$

Code:

For the MACD, we will need the `close` column, so grab that and then apply the `transform` method along with the specified Lambda function. Now calculating an Exponential Moving Average in `pandas` is easy. First, call the `ewm` (exponential moving weight) function and then specify the `span` or, in other words, the number of periods to look back. In this case, we use the definition provided by the formula and specify 26 & 12.

Once we've calculated the EMA_26 and EMA_12, we take the difference between EMA_12 & EMA_26 to get our MACD. Now that we have our MACD, we need to calculate the EMA of the MACD, so we take our MACD series and apply the same `ewm` function too, but in this case, we specify a `span` of 9. Finally, we add both the MACD and MACD_EMA to the main data frame.

In [68]:
```
# Calculate the MACD
ema_26 = price_data.groupby('symbol')['close'].transform(lambda x: x.ewm(span = 26).mean())
```

Activities Chromium Web Browser Oct 1 17:47
Math590_project/random_forest_price_prediction.ipynb at master · mukoedo1993/Math590_project - Chromium
Math590_project/random relative strength index - C Relative strength index - W
github.com/mukoedo1993/Math590_project/blob/master/random_forest_price_prediction.ipynb

Indicator Calculation: Price Rate Of Change

Definition From Paper:

It measures the most recent change in price with respect to the price in n days ago.

Formula:

$$PROC_t = \frac{C_t - C_{t-n}}{C_{t-n}} \quad (1)$$

(2)

(3)

where, (1)

$PROC_t$ = Price Rate of Change at time t (1)

C_t = Closing price at time t (1)

Code:

The Price Rate of Change is another easy indicator to calculate in pandas because we can leverage a built-in function. In this case, we will use the `pct_change` function and apply it to our all too familiar symbol groups. For the `pct_change` function, we have an argument called `periods` which specifies how far we need to look back when calculating the rate of change. In this case, the paper never provided a specific `n`, but after doing some research, I landed on an n of 9 because this seemed to be the standard window. Now, it's important to note that the paper changes n depending on the window, so technically I'm not doing exactly like they did. For example, if my prediction window was `30 days` then n should be 30.

In [69]:
```
# Calculate the Price Rate of Change
n = 9

# Calculate the Rate of Change in the Price, and store it in the Data Frame.
price_data['Price_Rate_Of_Change'] = price_data.groupby('symbol')['close'].transform(lambda x: x.pct_change(periods = n))

# Print the first 30 rows
```

Indicator Calculation: On Balance Volume

Definition From Paper:

On balance volume (OBV) (Granville 1976) utilizes changes in volume to estimate changes in stock prices. This technical indicator is used to d buying and selling trends of a stock, by considering the cumulative volume: it cumulatively adds the volumes on days when the prices group, and subtracts the volume on the days when prices go down, compared to the prices of the previous day.

Formula:

$$OBV(t) = \begin{cases} \text{OBV(t - 1) + Vol(t) if C(t) > C(t-1)} \\ \text{OBV(t - 1) - Vol(t) if C(t) < C(t-1)} \\ \text{OBV(t - 1) if C(t) = C(t-1)} \end{cases} \quad (1)$$

(2)

where, (3)

(4)

$\text{OBV (t)} = \text{on balance volume at time t}$ (5)

(6)

$\text{Vol(t)} = \text{trading volume at time t}$ (7)

(8)

$\text{C(t)} = \text{closing price at time t}$

Code:

This portion is a little more complicated than the previous ones. However, the idea is still the same. I'm going to be working with groups but in this case I'll be using the `apply` method to apply a custom function I built to calculate the `On Balance Volume`. The function simply calculates the `diff` for the closing price and uses a `for loop` to loop through each row in the `volume` column. If the `change in price` was greater than 0 we add the volume, if it's less than 0 we subtract the volume and if it's 0 then we leave it alone.

My work:

github/mlukovedo1993