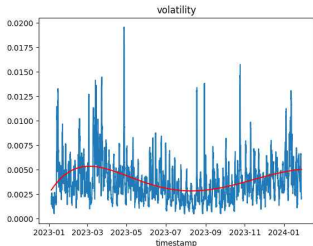






1. EDA



전역적 해석

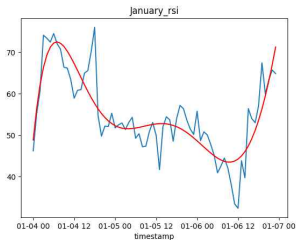
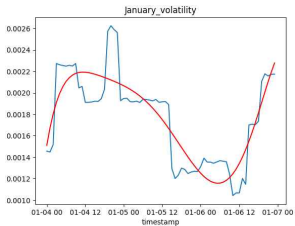
1. 시간에 따라 진동하는 양상을 보인다.
2. 연말 혹은 연초에 상승하는 모습을 보인다.
3. 3월에서 9월에는 하락하는 모습을 보인다.

->

- 휴가 시즌 혹은 연말 정산으로 인한 비트코인 투자
- 규제 및 정책 변화



비트코인은 시간에 따른 의존성을 갖고 있다.



국소적 해석

가장 흔한 주식 보조 지표 'rsi'를 비교해봅시다.

rsi의 양상과 volatility의 양상이 비슷하게 흘러감을 알 수 있음
rsi의 변화에 따른 사람들의 투자심리를 알 수 있음



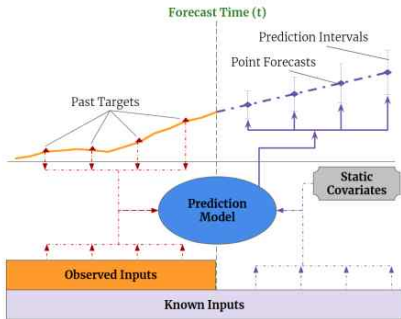
비트코인은 외부 요인 (보조 지표) 들의 영향을 받는다.

An orange oval shape serves as a background for the section header text.

2. Model Outline

Time-series forecasting model key idea

1. 설명 변수들과 변동성 간의 관계성



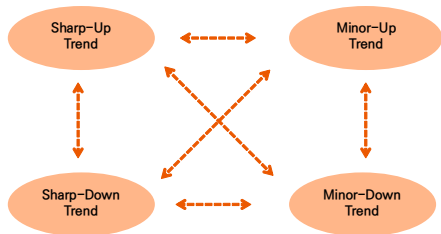
적절한 모델을 바탕으로 설명 변수들을 잘 생성할 수 있다면, 미래 변동성에 대해 잘 예측할 수 있을 것이다.



Regression 모델을 통해서 설명 변수와 변동성 간의 관계성을 포착하자

Time-series forecasting model key idea

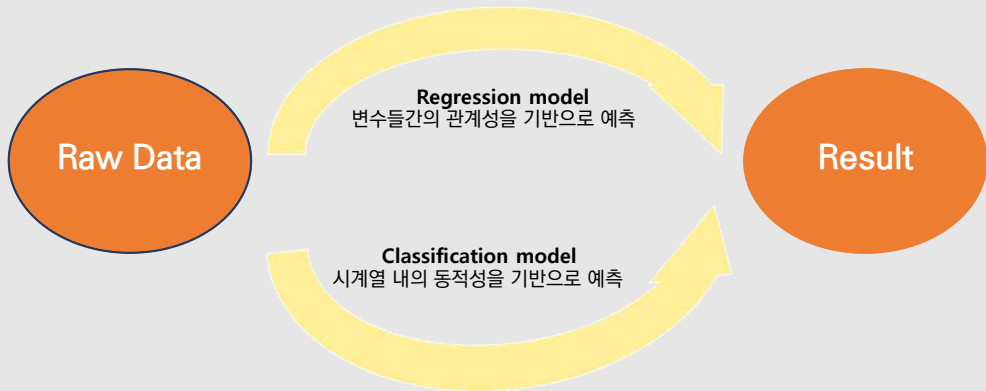
2. 시간에 따른 변동성의 동적성 포착



정확한 값이 아닌 연말 연초에 변동성이 커지는 흐름을 잡고자 변동의 유형을 4가지로 분류하고 유형을 예측하고자 하였다.



Classification 모델을 통해 변동성의 동적성 포착





3. Data Preprocessing & Feature Engineering

결측치 처리

open 값이
NaN인 경우

이전 시점의 Close 값을
Open 값으로 대체

open 외의
high, low,
close 값이
NaN인 경우

같은 시점의 Open 값으
로 대체

Volatility,
return이
NaN인 경우

각 열의 평균값으로 대
체



Scaling

산출 방식에 따라 다양
한 단위가 존재



Standard Scaler
사용!

feature 생성 방법	feature
Close 이용	MACD, rsi, TRIX, Momentum, Awesome_Oscillator, parabol
Open, High, Low , 이용	atr, williumR, adx, %K, %D, Aroon_Oscillator, AC, BOP, cci, plusdi, plusdm
거래량 이용	FI, MFI
HAR-RV 모형 이용	rv, rv_h, rv_d, rv_w
maker/taker 이용	taker/maker_ratio, taker-maker_diffsum

HAR-RV (Realize Volatility 예측을 위한 이질적자기회귀 모형)

	rv	rv_h	rv_d	rv_w
15	2.298141e-06	2.863043e-09	NaN	NaN
16	6.215545e-08	2.298141e-06	NaN	NaN
17	4.240558e-06	6.215545e-08	NaN	NaN
18	1.123596e-05	4.240558e-06	NaN	NaN
19	3.921944e-05	1.123596e-05	NaN	NaN
20	4.248017e-06	3.921944e-05	NaN	NaN
21	1.173443e-06	4.248017e-06	NaN	NaN
22	1.732108e-09	1.173443e-06	NaN	NaN
23	1.935739e-06	1.732108e-09	NaN	NaN
24	1.737455e-06	1.935739e-06	0.000006	NaN
25	9.100700e-08	1.737455e-06	0.000006	NaN

$$RV_{t+1} = \beta_0 + \beta_D RV_t + \beta_W RV_{t-5,t} + \beta_M RV_{t-22,t} + \epsilon_{t+1}$$

HAR-RV의 식

*시간별, 일별, 주별 rv평균을 사용한 HAR-RV 모델을 활용하여 예측 성능을 높이하고자 하였습니다.

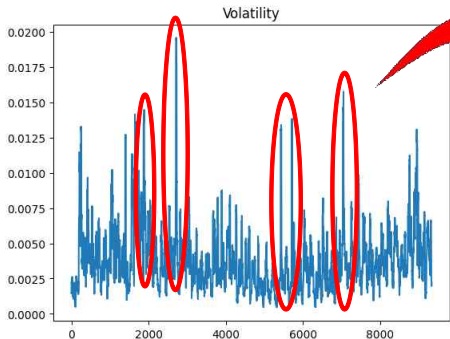
(Bouri, Elie, et al. "Forecasting realized volatility of bitcoin: The role of the trade war". 2021)

하루 전날의 평균

?

이전 일주일간의 평균

Maker&Taker

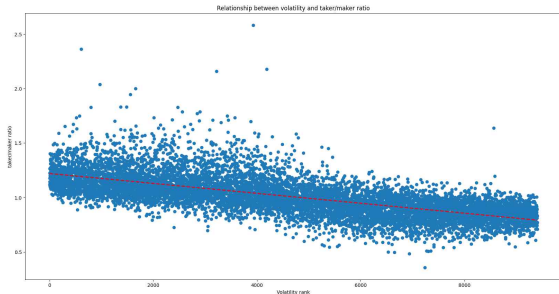


변동성이 큰 상승에선
taker 거래량
>>maker거래량



taker와 maker의 상대적인
비율을 변수로 고려

Maker&Taker



*가로축 : volatility의 rank
(volatility를 내림차순으로 정렬)

세로축 : taker/maker ratio

실제로 volatility가 클 때
taker/maker ratio 값이 큰
것을 확인!!

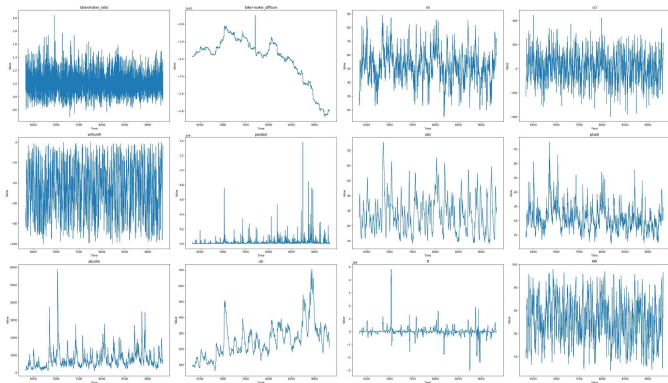
* taker-maker diffsum 변수

taker 비중이 크다고 무조건 변동성이 커지는 것이 아님+ 일반적으로 maker 거래 비중이 큰 편

-> 해당 변수를 통해 유동성 주체별 상대적인 거래대금 차이를 확인하고자 “누적합 변수”를 이용

An orange oval shape serves as a background for the section header text.

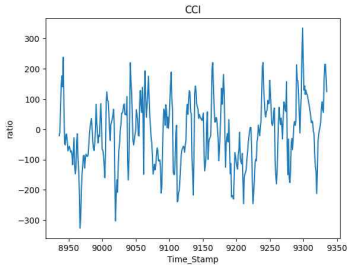
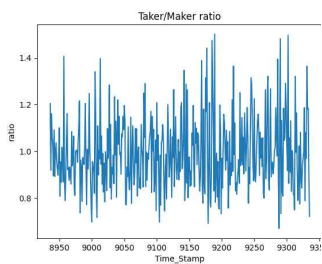
4. Regression Model



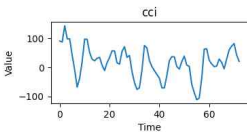
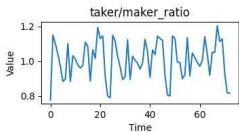
시간에 따른 주기적인 패턴을
확인 가능



각 시점을 독립적으로 보지 않고,
종속적인 시계열 데이터라고 판
단



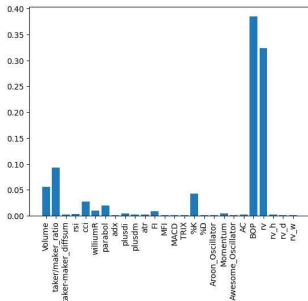
원 데이터



예측값

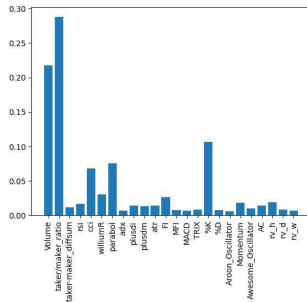
Validation 결과 치팅 요소가 있음을 확인

(특정 두 변수-BOP, rv-가 대부분의 importance를 차지하며, 그렇게 만들어진 모델의 성능이 좋지 않음) → 치팅요소를 제외하여 성능 개선



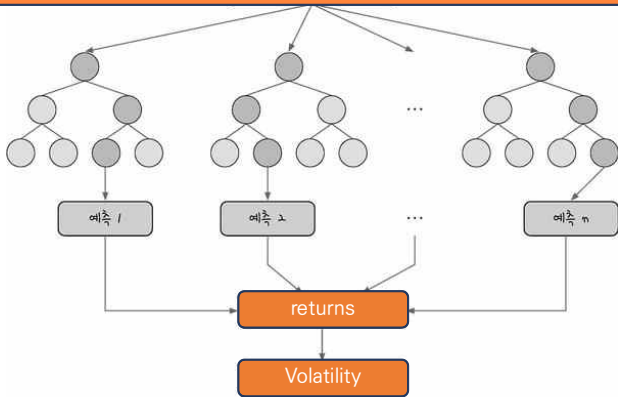
MAPE : 약 0.458

치팅요소 제거

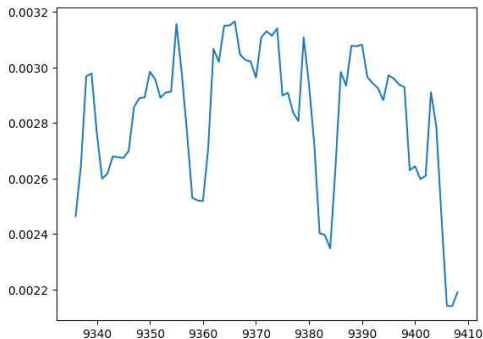
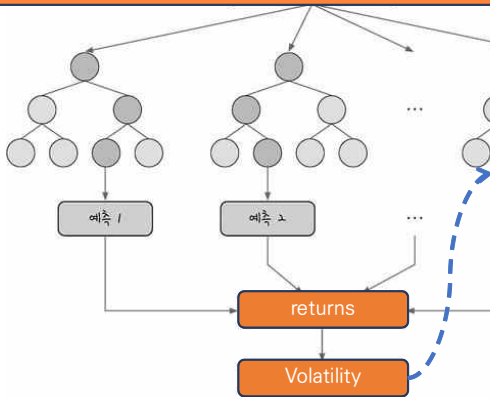


MAPE : 약 0.24로 감소!!

O,H,L,C,V로 생성한 feature / HAR-RV feature / Taker&Maker feature
들의 예측값



O,H,L,C,V로 생성한 feature / HAR-RV feature / Taker&Maker feature
들의 예측값





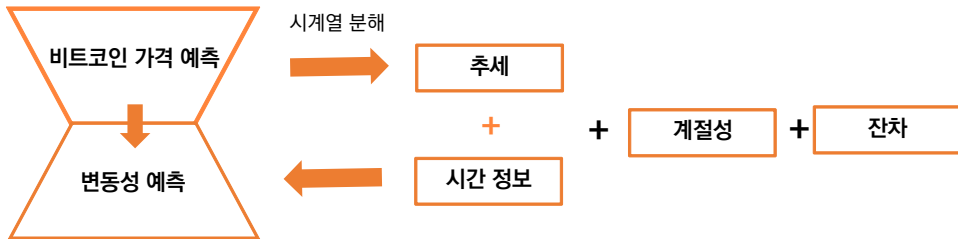
5. Classification Model

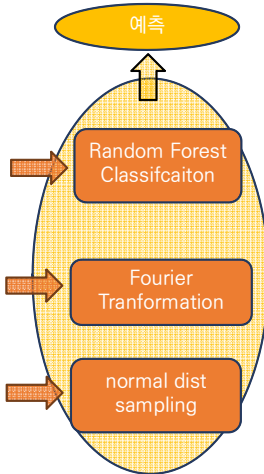
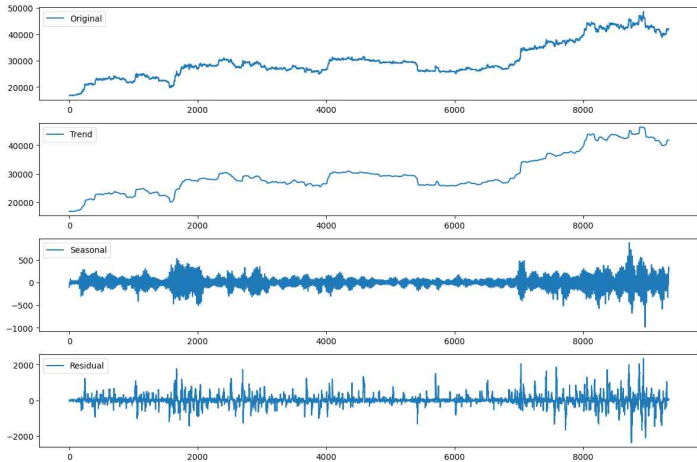
1. 동적성 포착

- > 시계열 데이터를 추세, 계절성, 잔차로 분해하여 추세를 통해 동적성 포착
- > 추세 변화량을 4가지 상태로 나누어서 상태 변화에 대한 예측

2. 시간 의존성 포착

- > 분류 모델에 시간 정보를 추가적으로 기입





Feature Engineering(1) : 날짜/시간

시간 의존성을 잡기 위해 텍데이터의 월, 일, 요일, 시간 값을 각각 one-hot-encoding으로 추가하여 컬럼 생성

month_4 month_5 month_6 month_7 month_8 month_9 month_10 ... hour_23 weekday_0 weekday_1 weekday_2 weekday_3

False	False	False	False	False	False	False	...	False	False	False	True	False
False	False	False	False	False	False	False	...	False	False	False	True	False
False	False	False	False	False	False	False	...	False	False	False	True	False
False	False	False	False	False	False	False	...	False	False	False	True	False
False	False	False	False	False	False	False	...	False	False	False	True	False

Feature Engineering(2) : trend 상승 정도를 나타내는 'class' + 'lag'

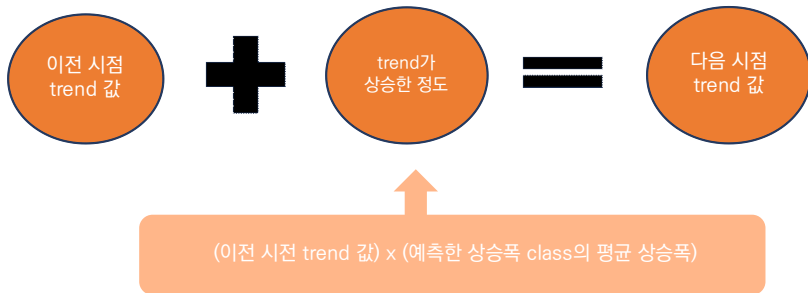
Trend 상승 정도(per) = (현재 시점의 trend - 이전 시점의 trend)/(현재 trend 값)

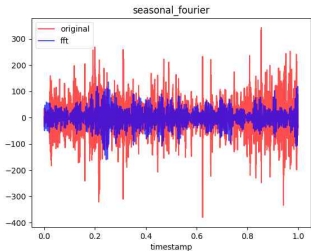
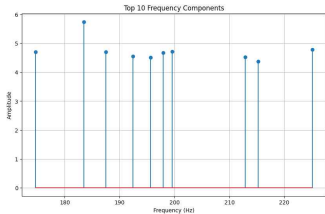
-> 이렇게 구한 trend 상승 정도(변수명: 'per')를 quantile로 나눠 각각 1~4의 class를 부여(변수명: 'class')

-> 이전 7시점의 'class'(상승정도)를 컬럼으로 추가해줌으로써 시간간의 관계성 정보 추가

trend	per	class		class	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7
16816.690641	0.000046	3		2023-03-28 05:00:00	1	1.0	1.0	2.0	2.0	2.0	2.0
22052.530226	0.001235	4		2023-03-28 06:00:00	1	1.0	1.0	1.0	2.0	2.0	2.0
26946.386893	-0.000117	2		2023-03-28 07:00:00	1	1.0	1.0	1.0	2.0	2.0	2.0
27581.584953	0.000008	3		2023-03-28 08:00:00	2	2.0	1.0	1.0	1.0	2.0	2.0
26593.094817	0.001118	4		2023-03-28 09:00:00	2	2.0	2.0	1.0	1.0	1.0	2.0
29367.232744	-0.000363	2		2023-03-28 10:00:00	3	2.0	2.0	2.0	1.0	1.0	1.0
25779.994888	0.000037	3		2023-03-28 11:00:00	3	3.0	2.0	2.0	2.0	1.0	1.0
29896.753961	0.000492	4		2023-03-28 12:00:00	4	3.0	3.0	2.0	2.0	2.0	1.0
39474.596917	0.000399	3		2023-03-28 13:00:00	4	4.0	3.0	2.0	2.0	2.0	1.0
				2023-03-28 14:00:00	4	4.0	4.0	3.0	3.0	2.0	2.0

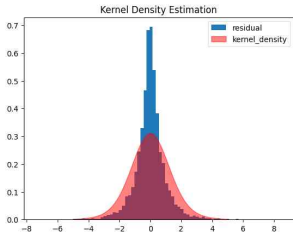
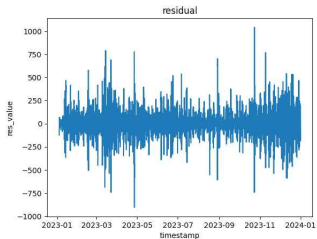
lag 변수 추가





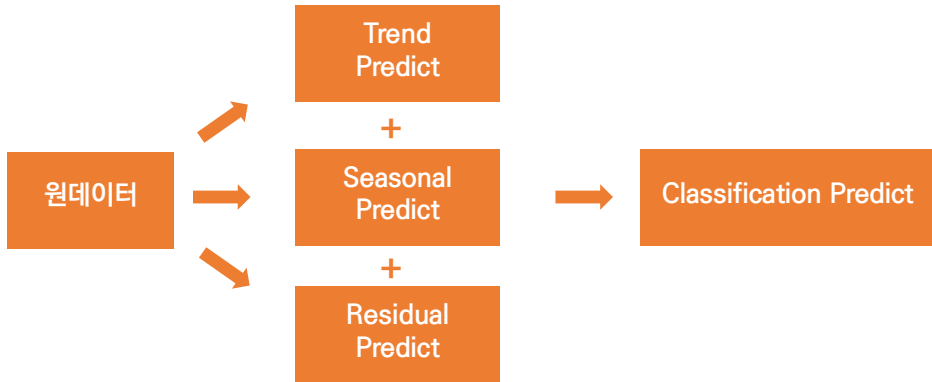
계절성, 주기성을 갖는값

- 주기성을 갖는 값들은 푸리에 변환을 통해 주파수 분해
- 분해된 주파수 중 500개의 주파수를 이용하여 다음 시점 값 예측



잔차, 설명되지 않는 값

- 잔차의 그래프를 그렸을 때 어느 정도 동일 분포 내에서 값이 나온다고 판단
- 정규화를 시켜준 이후, 가우시안 커널을 통해 샘플링하는 방식으로 진행
- 샘플링한 값을 다시 복원해주어 잔차 예측값으로 사용







감사합니다