결측치 조사_0127

CH1. Model Explanation

1. 기본 보간법

기본 보간법은 결측치를 주변의 관측치의 평균, 중앙값, 최빈값 등으로 대체하는 방법이다. 이 방법은 간단하고 직관적이지만, 시계열 데이터의 특성을 고려하지 않고 결측치를 대체하므로 정확한 예측을 위해서는 한계가 있다. 주로 데이터가 일정한 주기로 변동하지 않는 경우에 사용된다.

→ 우리 데이터의 결측치를 처리하는 데에는 적절하지 않을 방법이라 생각됨

2. 스플라인

스플라인은 부드러운 곡선을 형성하여 결측치를 대체하는 방법이다. 주변의 관측치를 이용하여 보간함수를 구성하고, 이를 통해 결측치를 추정한다. 스플라인은 시계열 데이터의 비선형적인 패턴을 고려하여 대체하기 때문에 비교적 정확한 예측을 할 수 있다.

3. ARIMA (자동회귀이동평균)

ARIMA 모델은 자동회귀와 이동평균을 결합한 모델로, 시계열 데이터의 추세와 계절성을 고려하여 예측한다. ARIMA 모델은 결측치 처리에 용이한 모델 중 하나이다. 결측치가 있는 시계열 데이터를 ARIMA 모델에 적용할 때, 알려진 데이터를 활용하여 모델을 학습하고, 이를 통해 결측치를 예측한다.

4. GARCH (일반화 자기회귀 조건부 헤스티모델)

GARCH 모델은 시계열 데이터의 변동성을 모델링하는데 사용되는 모델이다. GARCH 모델은 결측치 처리에 용이한 모델 중 하나로 알려져 있다. GARCH 모델은 시계열 데이터의 변동성을 추정하고, 이를 통해 결측치를 예측한다. GARCH 모델은 특히 금융 시계열 데이터에서 사용되며, 변동성 예측에 뛰어난 성능을 보인다.

☞ 이렇게 기본 보간법, 스플라인, ARIMA, GARCH 모델은 각자의 특성에 따라 시계열 데이터 결측치 처리에 활용된다.

CH2. Model Pros/Cons

1. 기본 보간법

장점:

- 간단하고 직관적인 방법으로 결측치 대체

- 데이터의 특성을 고려하지 않고 결측치를 대체하기 때문에, 데이터가 일정한 주기로 변동하는 경우에 유용

단점:

- 데이터의 패턴과 특성을 고려하지 않기 때문에, 정확한 예측을 위해서는 한계가 존재
- 대체된 결측치의 예측력이 낮을 수 있음

2. 스플라인

장점:

- 시계열 데이터의 비선형적인 패턴을 고려하여 대체하기 때문에, 비교적 정확한 예측 가능
- 스플라인은 보간 함수를 사용하여 결측치를 추정하기 때문에, 부드러운 곡선 형성 가능

단점:

- 스플라인 모델은 주변의 관측치에 의존하기 때문에, 관측치의 간격이 크거나 데이터의 특성이 변동성이 큰 경우에는 예측력이 떨어질 수 있음
- 모델의 계산량이 많을 수 있어, 대규모 데이터셋에서는 시간이 오래 걸릴 수 있음

3. ARIMA

장점:

- 자동회귀와 이동평균을 결합한 모델로, 시계열 데이터의 추세와 계절성을 고려하여 예측
- 결측치가 있는 시계열 데이터를 ARIMA 모델에 적용할 때, 알려진 데이터를 활용하여 모델을 학습하고, 이를 통해 결측치를 예측 가능

단점:

- ARIMA 모델은 데이터의 정상성을 가정하고 있기 때문에, 비정상적인 데이터에는 적용하기 어려움
- ARIMA 모델은 모수 추정과 모델 선택 과정에서 주관적인 판단이 필요

4. GARCH

장점:

- 시계열 데이터의 변동성을 모델링하는데 사용되어, 변동성 예측에 뛰어난 성능
- GARCH 모델은 금융 시계열 데이터에서 특히 많이 사용되며, 변동성 예측에 유용

단점:

- GARCH 모델은 모수 추정이 복잡하고, 계산량이 많을 수 있음
- GARCH 모델은 변동성에만 초점을 맞추기 때문에, 시계열 데이터의 다른 특성에 대한 정보 미제공

☞ 각 모델은 자체적인 장단점을 가지고 있으며, 데이터의 특성과 분석 목적에 따라 적절한 모델을 선택하면 된다.

CH3. Other Method

1. 결측치 제거

- 결측치가 있는 행 또는 열을 제거하는 방법
- 데이터셋이 충분히 크고 결측치가 일부일 경우에 사용할 수 있다. 하지만 결측치가 다수의 행 또는 열에 걸쳐 존재한다면, 데이터의 손실이 크게 발생할 수 있으므로 신중하게 사용해야 한다.

2. 상수로 대체

- 결측치를 특정 상수 값으로 대체하는 방법
- 예를 들어, 결측치를 0 으로 대체하거나, 평균값으로 대체하는 등의 방법이 있다. 이 방법은 결측치의 정보 손실이 발생할 수 있으므로 주의가 필요하다.

3. 예측 모델 활용

- 다른 변수들을 활용하여 결측치를 예측하는 모델을 구축하는 방법
- 예를 들어, 다중 선형 회귀 모델이나 K-NN(K-Nearest Neighbors) 등의 알고리즘을 사용하여 결측치를 예측하는 방법이 있다. 이 방법은 데이터의 패턴을 더 잘 반영할 수 있지만, 모델 구축에 시간과 노력이 필요하다.

```
# 결측치 처리: K-NN 방법
knn_imputer = KNNImputer(n_neighbors=3)
df_knn_imputed = pd.DataFrame(knn_imputer.fit_transform(df),
columns=df.columns)
```

4. 다중 대치법(Multiple Imputation)

- 결측치를 예측하기 위해 여러 번의 대치를 수행하는 방법
- 다중 대치법은 결측치를 예측하는 모델을 여러 번 실행하여 여러 개의 예측값을 생성한 후, 이를 평균 또는 중간값으로 결합하여 최종 예측값을 구하는 방법
- 이를 통해 불확실성을 반영할 수 있으며, 결측치 처리의 정확성을 향상시킬 수 있다.

☞ 이러한 방법들은 데이터의 특성과 결측치의 패턴에 따라 선택되어야 한다. 데이터셋의 크기, 결측치의 비율, 분석 목적 등을 고려하여 적절한 결측치 처리 방법을 선택해야 한다.

CH4. Interpolation Cons

1. 결측치가 큰 비율로 존재하는 경우

- 결측치가 데이터의 큰 비율을 차지할 경우, 보간법은 대체로 부적합할 수 있다.

- 이는 보간법이 결측치를 주변 값으로 대체하기 때문에, 결측치가 많은 경우에는 데이터의 왜곡이 심해질 수 있다. 이런 경우에는 결측치를 대체하기보다는, 결측치를 제거하거나 다른 방법을 고려하는 것이 더 적합하다.

2. 결측치가 패턴을 가지고 있는 경우
- 결측치가 특정한 패턴을 가지고 있다면, 보간법은 해당 패턴을 정확하게 반영하기 어렵다.
- 예를 들어, 요일별로 주기성을 가진 데이터에서 특정 요일의 값이 결측치인 경우, 해당 요일의 평균값으로 대체하는 것보다는 해당 요일의 평균값을 고려한 다른 방법을 사용하는 것이 더 적합하다.

3. 데이터 간의 관계가 복잡한 경우
- 보간법은 결측치를 주변 값으로 대체하는 방식이므로, 데이터 간의 복잡한 관계를 정확하게 반영하기 어렵다.
- 이런 경우에는 예측 모델을 사용하여 결측치를 예측하는 방법이 더 적합할 수 있다. 예를 들어, 다중 선형 회귀 모델이나 시계열 모델을 사용하여 결측치를 예측하고 대체할 수 있다.

→ 따라서 결측치 처리 방법을 선택할 때는 데이터의 특성과 결측치의 패턴을 고려해야 한다. 보간법은 간편하고 빠른 방법이지만, 데이터 왜곡이나 패턴 무시 등의 문제가 발생할 수 있으므로, 상황에 맞는 다른 방법을 고려하는 것이 중요하다.

CH5. Time-series Data-NaN processing

1. 선형 보간 (Linear Interpolation)
- 시계열 데이터에서 인접한 데이터 포인트 사이의 직선을 사용하여 결측치를 보간
- 이 방법은 데이터의 추세를 유지하면서 결측치를 대체

2. 시간에 따른 평균값 대체 (Time-based Mean Imputation):
- 시간에 따라 그룹화하여 해당 시간 범위의 평균값으로 결측치를 대체
- 시간에 따른 패턴을 고려하여 결측치를 보완

3. 이전 값으로 대체 (Forward Fill)
- 이전의 유효한 값으로 결측치를 대체
- 시계열 데이터에서의 지속적인 값의 변화가 크지 않을 때 유용

4. 다음 값으로 대체 (Backward Fill)
- 다음 유효한 값으로 결측치를 대체하는 방법

- 시계열 데이터에서 값의 변화가 크지 않을 때 적합

5. 시계열 예측 모델 활용
- 결측치를 예측하는 시계열 모델을 구축하여 결측치를 대체하는 방법
- ARIMA, Prophet, LSTM과 같은 모델을 사용하여 결측치를 예측

☞ 위의 방법들을 활용하여 결측치를 처리할 수 있다. 하지만 어떤 방법이 가장 적합한지 판단하기 위해서는 데이터의 특성과 분석 목적을 고려해야 한다. 또한, 결측치를 대체한 후에는 데이터의 왜곡이나 영향력을 확인하기 위해 추가적인 분석을 수행하는 것이 좋다.

☞ 선형 보간은 주변의 데이터를 활용하여 결측치를 추정하는 방법으로, 데이터의 추이를 고려한 보간 결과를 얻을 수 있다. 반면에 평균값 대체는 결측치를 해당 열의 평균값으로 대체하는 방법으로, 단순하면서도 효과적인 방법이다. 선형 보간은 데이터의 연속성을 고려하고자 할 때 유용하며, 평균값 대체는 결측치를 대체하는 간단한 방법이다.

CH6. Method Suitability

어떤 결측치 처리 방법이 우리 데이터의 컬럼에 가장 적합한가

1. 데이터의 패턴과 추세:
- 데이터가 시간에 따라 어떤 패턴을 가지고 있는지 확인
- 추세, 계절성, 주기성 등의 패턴이 있는 경우 선형 보간 또는 시간에 따른 평균값 대체가 적합할 수 있다.

2. 결측치의 위치와 분포:
- 결측치가 어디에 위치하고 있는지, 그리고 결측치의 분포가 어떻게 되는지 확인
- 만약 결측치가 연속적으로 나타나는 경우에는 선형 보간이나 시계열 예측 모델을 활용하는 것이 적합할 수 있다.

3. 데이터의 왜곡 여부:
- 결측치를 대체한 후에도 데이터의 왜곡이 발생하는지 확인
- 대체한 값이 원래 데이터와 일치하지 않거나, 데이터의 분포에 큰 변화를 주는 경우 다른 방법을 고려

☞ 결측치를 대체한 후에는 데이터의 왜곡이나 영향력을 확인하기 위해 추가적인 분석을 수행하는 것이 좋다.

CH7. KAGGLE

1) https://www.kaggle.com/code/ysthehurricane/bitcoin-dogecoin-etc-price-prediction-xgboost#Introduction

Preprocessing

Replace None or Null values with forward filling technique using *fillna* function.

In [64]:
```
bitcoindf = bitcoindf.fillna(method = 'ffill')
dogecoindf = dogecoindf.fillna(method = 'ffill')
ethereumdf = ethereumdf.fillna(method = 'ffill')
cardanodf = cardanodf.fillna(method = 'ffill')
```

ffill method로 결측치 처리

2) https://www.kaggle.com/code/meetnagadia/xgboost-bitcoin-price-prediction#5.-Checking-for-null-values

5. Checking for null values

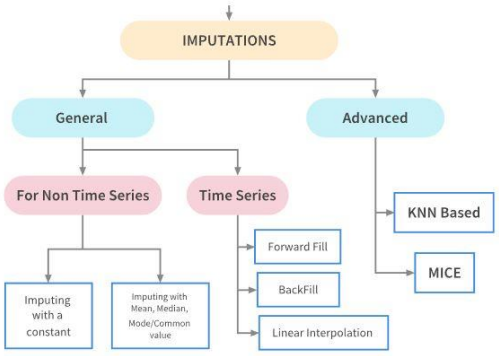
In [7]:
```
data.isnull().sum()
```

Out[7]:
```
date         0
open         4
high         4
low          4
close        4
adj_close    4
volume       4
dtype: int64
```

2534개의 행 중에 4개의 결측치 무시하고 진행

3) https://www.kaggle.com/code/akashmathur2212/bitcoin-price-prediction-arima-xgboost-lstm-fbprop#Handling-Missing-Values-in-Time-series-Data

	Total Missing Values	Missing %
Timestamp	0	0.000000
Open	1241716	27.157616
High	1241716	27.157616
Low	1241716	27.157616
Close	1241716	27.157616
Volume_(BTC)	1241716	27.157616
Volume_(Currency)	1241716	27.157616
Weighted_Price	1241716	27.157616



Imputation using Linear Interpolation method

Time series data has a lot of variations against time. Hence, imputing using backfill and forward fill isn't the best possible solution to address the missing value problem. A more apt alternative would be to use interpolation methods, where the values are filled with incrementing or decrementing values.

Linear interpolation is an imputation technique that assumes a linear relationship between data points and utilises non-missing values from adjacent data points to compute a value for a missing data point.

Refer to the official documentation for a complete list of interpolation strategies here

In our dataset, we will be performing Linear interpolation on the missing value columns.

☞ 시계열 데이터는 시간에 따라 변수 多 점을 고려해 선형 보간법으로 전체 결측치 대체

4) https://www.kaggle.com/code/adityamhaske/bitcoin-price-prediction#Handling-Missing-Values

Handling Missing Values

In [10]:

```
# Handling missing values in df_daily
df_daily = df_daily.fillna(df_daily.mean())

# Handling missing values in df_monthly
df_monthly = df_monthly.fillna(df_monthly.mean())

# Handling missing values in df_annual
df_annual = df_annual.fillna(df_annual.mean())

# Handling missing values in df_quarterly
df_quarterly = df_quarterly.fillna(df_quarterly.mean())
```

평균으로 결측치 대체

5) https://www.kaggle.com/code/someadityamandal/bitcoin-time-series-forecasting

In [5]:

```
# First thing is to fix the data for bars/candles where there are no trades.
# Volume/trades are a single event so fill na's with zeroes for relevant fields...
data['Volume_(BTC)'].fillna(value=0, inplace=True)
data['Volume_(Currency)'].fillna(value=0, inplace=True)
data['Weighted_Price'].fillna(value=0, inplace=True)

# next we need to fix the OHLC (open high low close) data which is a continuous timeseries so
# lets fill forwards those values...
data['Open'].fillna(method='ffill', inplace=True)
data['High'].fillna(method='ffill', inplace=True)
data['Low'].fillna(method='ffill', inplace=True)
data['Close'].fillna(method='ffill', inplace=True)

data.head()
```

Out[5]:

	Timestamp	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price
0	2014-12-01 05:33:00+00:00	300.0	300.0	300.0	300.0	0.01	3.0	300.0
1	2014-12-01 05:34:00+00:00	300.0	300.0	300.0	300.0	0.00	0.0	0.0
2	2014-12-01 05:35:00+00:00	300.0	300.0	300.0	300.0	0.00	0.0	0.0
3	2014-12-01 05:36:00+00:00	300.0	300.0	300.0	300.0	0.00	0.0	0.0
4	2014-12-01 05:37:00+00:00	300.0	300.0	300.0	300.0	0.00	0.0	0.0

ffill method로 결측치 대체

추가) Q&A

1. 전처리 중 데이터에 결측치가 발생한 경우에 보간법 알고리즘을 사용해도 될까요?\

그리고 데이터 정보에서 컬럼명 오류 있습니다.

is_buy_market(X) → is_buyer_market(O)

답변

안녕하세요, 향기님

문의 주신 내용 답변 드립니다.

결측치와 결측치 처리에 관해 주최사에서 정한 규정과 방법이 없습니다.

소스 코드에 markdown으로 추가 설명 작성이 가능합니다.

→ 보간법 사용

개인 의견)

어떤 방법을 진행하느냐에 따라 결측치를 다루는 방식이 달라지고, 그 값이 달라지겠지만

우리가 원하는 대로, 이상적으로, 완벽하게 결측치를 보완할 수는 없을 것이다.

데이터에는 우리가 고려하지 않는 외부 요인과 사회, 뉴스 등이 분명히 영향을 끼칠 것인데

보간법, GARCH, ARIMA 등의 방법으로 그 결측치와 데이터를 100% 적합시키고 다룰 순 없다고 생각한다.

그리고 "이런 유형의 데이터에는 이 방법을 시도해볼 순 있어~"는 여럿 볼 수는 있지만, "이 컬럼에는 무조건 이 방법이 답이야!"는 알아낼 수 없으며 법칙 또한 존재하지 않는다.

여러 방법을 시도해보며, 조사해가며 결측치를 다루는 건 좋지만

데이터의 3%에 불과하는 결측치를 오랫동안 시간 끌 필요는 없을 것이라 생각한다. (아닐 수도...ㅎㅎ)

지금까지 우리가 시도해보고 조사해본 결측치 처리 방법 중에서는 "보간법"이 제일 금융 시계열 데이터에 적합해 보인다.

아자아자 파이팅~