

캡스톤 디자인 I 최종결과 보고서

프로젝트 제목(국문): 빅데이터 분석 기반 비트코인 가격 예측 모델

프로젝트 제목(영문): Bitcoin price prediction model based on big data analysis

프로젝트 팀(원): 학번: 20181590 이름: 김희섭
프로젝트 팀(원): 학번: 20181581 이름: 강현욱

1. 중간보고서의 검토결과 심사위원의 '수정 및 개선 의견'과 그러한 검토의견을 반영하여 개선한 부분을 명시하시오.

- 수정 및 개선 의견

1) 지표 별 AI 학습을 통한 정확도를 분석 시 딥러닝 모델로 학습시켜 검증하는 것이 아닌 지도학습 모델의 선형회귀 방식을 사용하는 것이 더 나을 수 있다고 하심

: Linear regression 방식을 통한 지표별 회귀 분석을 진행함.

2) 지표 별 회귀분석 시 데이터를 분할할 때 랜덤하게 보단 시계열 데이터이므로 순서를 지켜 분할하기를 권장하심

: train/test set 분할 함수 중 하나인 train_set_split 함수의 인수 중 shuffle을 False로 둬으로써 무작위로 값이 섞여 분할되지 않도록 함.

3) 분석 이후 바로 딥러닝 기반 모델을 만들기 보단 지도학습을 통한 비교 모델을 만들어 진행하는 방식을 추천해주심

: Linear regression 방식으로 모든 지표를 학습 데이터로 넣고, 다항 차수로 fit한 다음 Linear / Ridge / Lasso 방식으로 지도학습 진행하여 비교 모델을 만들었음.

2. 기능, 성능 및 품질 요구사항을 충족하기 위해 본 개발 프로젝트에서 적용한 주요 알고리즘, 설계방법 등을 기술하시오.

1) 예측을 위한 비트코인 가격 데이터 호출 : upbit api 를 통해 가격 데이터를 불러옴

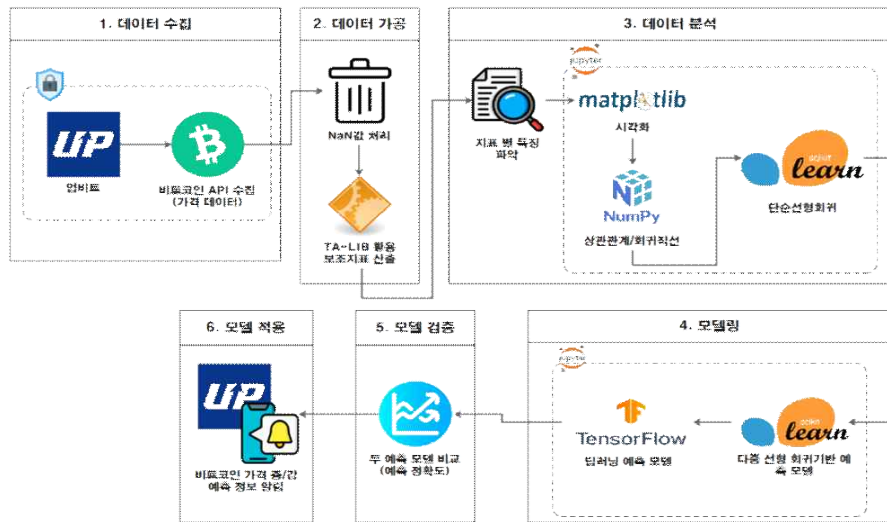
2) 보조지표 계산 : ta-lib 이라는 python library를 통해 대상 지표를 불러옴

3) 보조지표 데이터 분석 : 시각화 작업은 matplotlib을 통해 이루어짐, 상관분석은 numpy의 상관계수 계산 함수를 이용해 값을 계산 후, 이를 matplotlib을 통해 시각화함, 회귀분석은 sklearn에서 linear regression을 이용해 진행, 결과는 matplotlib을 통해 시각화

4) 모델 훈련 및 예측 : 딥러닝 기반 모델링은 tensorflow로 진행예정, 비교 모델의 모델링은 sklearn의 linear regression을 polynomial을 이용해 다항으로 변형하여 학습, input으론 모든 지표값을 넣었고 방식은 'linear', 'ridge', 'lasso' 방식 3가지로 진행

3. 요구사항 정의서에 명시된 기능 및 품질 요구사항에 대하여 최종 완료된 결과를 기술하시오.

- 구성도



1) 예측을 위한 비트코인 가격 데이터 호출

```
df = up.get_ohlcv(ticker="KRW-BTC", interval="day", count=2085)
```

	Unnamed: 0	open	high	low	close	volume	value
0	2017-09-25 09:00:00	4201000.0	4333000.0	4175000.0	4322000.0	132.484755	5.602146e+08
1	2017-09-26 09:00:00	4317000.0	4418000.0	4311000.0	4321000.0	22.788340	9.950724e+07
2	2017-09-27 09:00:00	4322000.0	4677000.0	4318000.0	4657000.0	32.269662	1.448276e+08
3	2017-09-28 09:00:00	4657000.0	4772000.0	4519000.0	4586000.0	80.588243	3.721860e+08
4	2017-09-29 09:00:00	4586000.0	4709000.0	4476000.0	4657000.0	59.352373	2.724558e+08
...
2080	2023-06-06 09:00:00	34299000.0	35993000.0	33580000.0	35852000.0	5402.995691	1.866340e+11
2081	2023-06-07 09:00:00	35855000.0	35993000.0	34797000.0	34952000.0	4520.054910	1.596646e+11
2082	2023-06-08 09:00:00	34953000.0	35416000.0	34802000.0	34996000.0	2370.319951	8.300358e+10
2083	2023-06-09 09:00:00	34996000.0	35161000.0	34802000.0	34865000.0	2335.692703	8.164586e+10
2084	2023-06-10 09:00:00	34865000.0	34914000.0	33842000.0	34386000.0	5968.685701	2.043161e+11

2085 rows x 7 columns

2) 보조지표 계산

```
# 이동평균
df_data['MAS'] = ta.SMA(df['close'], timeperiod=5)
df_data['MA20'] = ta.SMA(df['close'], timeperiod=20)

# RSI
df_data['RSI'] = ta.RSI(df['close'], timeperiod=14)

# MACD
df_data['MACD'], df_data['Signal'], df_data['Hist'] = ta.MACD(df['close'], fastperiod=12,
                                                             slowperiod=26, signalperiod=9)

# 볼린지 밴드
df_data['Upper'], df_data['Middle'], df_data['Lower'] = ta.BBANDS(df['close'], timeperiod=20,
                                                                nbdevup=2, nbdevdn=2)

# 스토캐스틱
df_data['SlowK'], df_data['SlowD'] = ta.STOCH(df['high'], df['low'], df['close'],
                                              fastk_period=5, slowk_period=3, slowd_period=3)

# OBV
df_data['OBV'] = ta.OBV(df['close'], df['volume'])

# ADX
df_data['ADX'] = ta.ADX(df['high'], df['low'], df['close'])

# ATR
df_data['ATR'] = ta.ATR(df['high'], df['low'], df['close'])

# WFI
df_data['WFI'] = ta.WFI(df['high'], df['low'], df['close'], df['volume'], timeperiod=14)

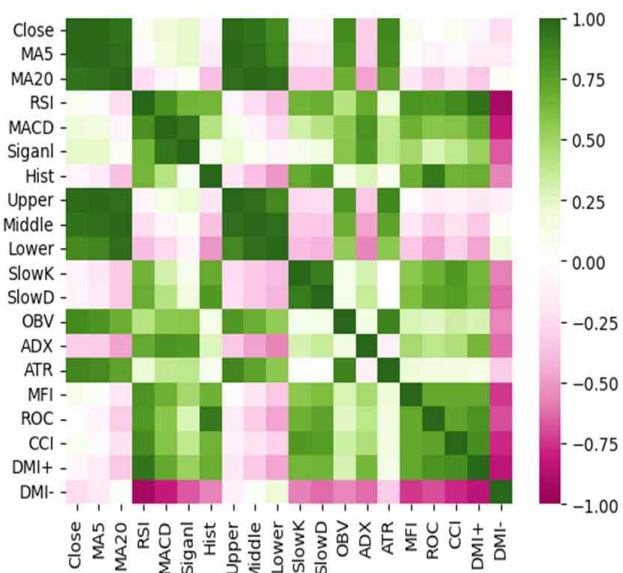
# ROC
df_data['ROC'] = ta.ROC(df['close'], timeperiod=10)

# CCI
df_data['CCI'] = ta.CCI(df['high'], df['low'], df['close'], timeperiod=20)

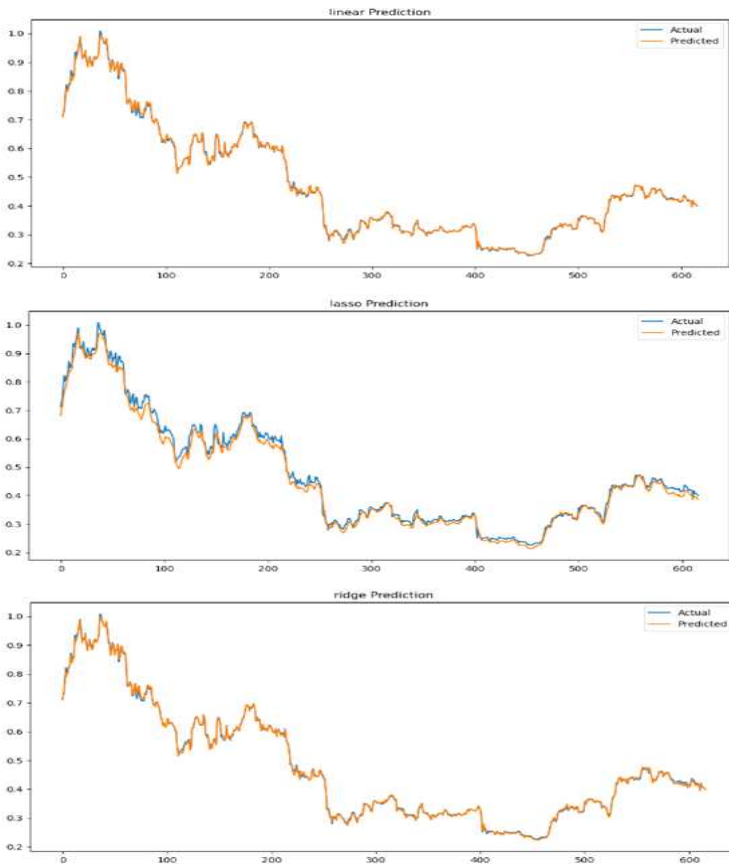
# DMI
df_data['DM+'] = ta.PLUS_DI(df['high'], df['low'], df['close'], timeperiod=14)
df_data['DM-'] = ta.MINUS_DI(df['high'], df['low'], df['close'], timeperiod=14)
```

3) 보조지표 데이터 분석

	이동평균(MA5/MA20)	RSI	MACD(MACD/Signal/Hist)	볼린저 밴드 (Upper/Middle/Lower)
사용변수	Close	Close	Close	Close
시각화				
상관분석 회귀직선				
회귀분석				
	스토캐스틱(SlowK/SlowD)	OBV	ADX	ATR
사용변수	High, Low, Close	Close, Volume	High, Low, Close	High, Low, Close
시각화				
상관분석 회귀직선				
회귀분석				
	MFI	ROC	CCI	DMI(+/-)
사용변수	High, Low, Close, Volume	Close	High, Low, Close	High, Low, Close
시각화				
상관분석 회귀직선				
회귀분석				



4) 비교 용 다중선형회귀 예측 모델



4. 구현하지 못한 기능 요구사항이 있다면 그 이유와 해결방안을 기술하시오,

최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등)
LSTM 모델 훈련 및 예측	미구현	해당 항목은 캡스톤 디자인 II 범위에 해당

5. 요구사항을 충족시키지 못한 성능, 품질 요구사항이 있다면 그 이유와 해결방안을 기술하시오.

분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등)
예측 정확도 및 소요 시간	미측정	앞선 모델 구현이 캡스톤 디자인 II 범위에 해당하여 모델이 미구 되었기 때문
실제 투자에서의 적합성	미측정	앞선 모델 구현이 캡스톤 디자인 II 범위에 해당하여 모델이 미구 되었기 때문

6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

모델 자체가 미구현이기 때문에 구체적인 매뉴얼은 없음

구현 이후에는 저장된 예측 모델을 호출하는 것만으로 사용 가능

7. 캡스톤디자인 결과의 활용방안

예측은 일별 데이터를 기준으로 예측, 다음날의 종가를 상승/하락폭에 따라 확률(%)로 알려줌

이에 따라 단순 예측 모델로 쓰일 수도 있고, 혹은 새로운 보조지표로도 활용될 수 있음, 업비트 api에서의 자동 매수/매도 시스템과 결합해 다음날 0시가 될 때 사용자에게 알림으로 예측값을 보여주고, 사용자는 원하는 매수/매도 범위를 선택, 이 범위에 가격이 도달할 경우 자동으로 매수/매도를 할 수 있도록 하는 자동화 시스템으로도 활용 가능 할 것으로 보인다.