

# Trading Bot

- ❖ 프로젝트 진행일 : 2019.01.07 ~ 2019.10.22
- ❖ 소속 회사명 : (주)뉴럴비씨
- ❖ 개발 환경 : Naver Cloud Platform Server (8 vCPUs, 60GiB Memory, 2 P40 GPUs, Ubuntu)
- ❖ 주사용 기술 : Machine Learning, Python, MySQL, MongoDB

## 프로젝트 개요

- ❖ AI Model이 암호화폐 매매에 도움을 줄 수 있도록 상승 또는 하락 징후를 포착하여 유저들에게 푸시 알림을 제공하는 서비스
- ❖ AI Model이 예측한 신호를 바탕으로 유저들이 수익을 볼 수 있도록 각 거래소에서의 자동 매매를 지원하는 서비스



Telegram

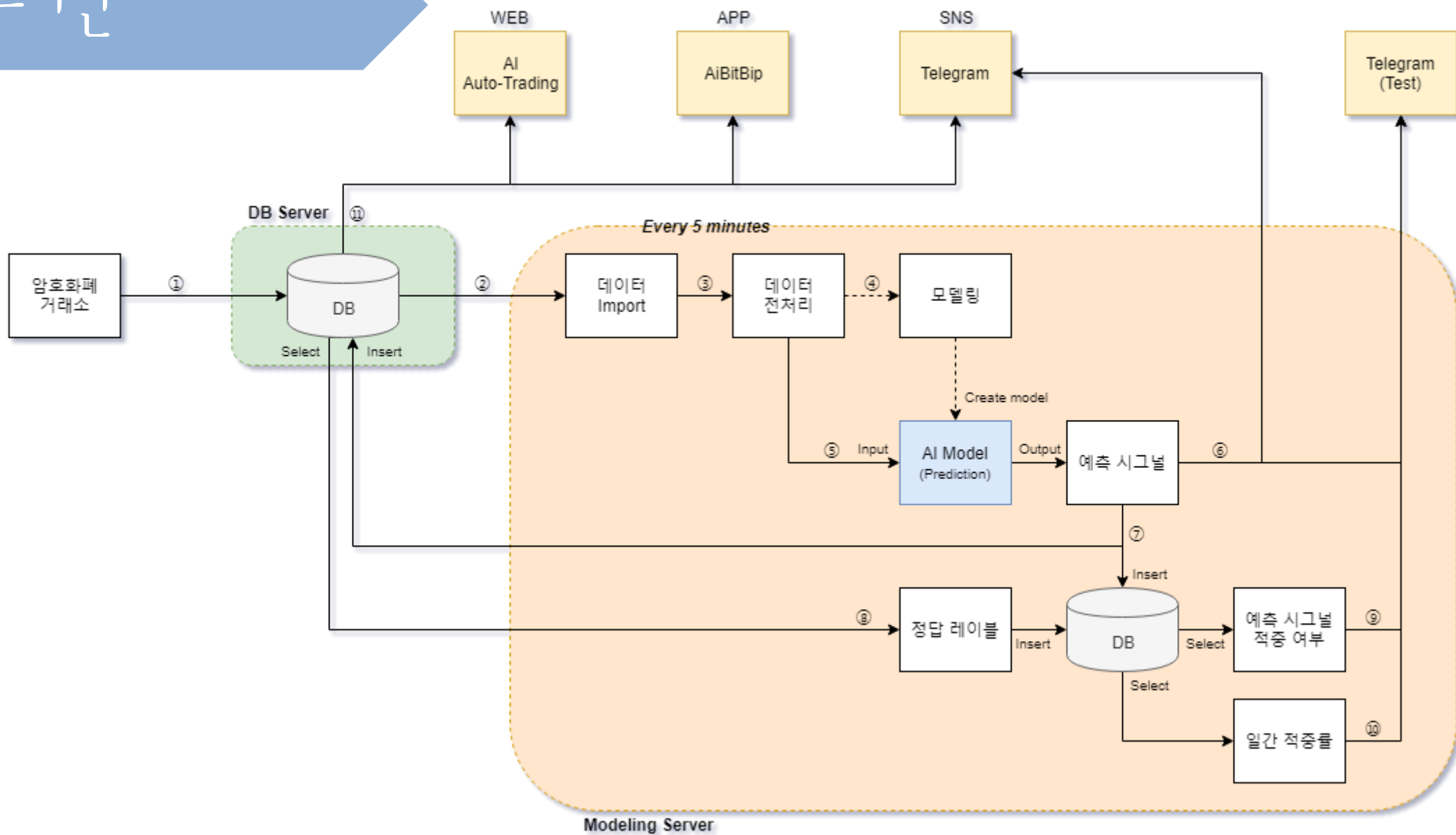


AiBitBip



AI Auto-Trading

# 파이프라인



## 요약

- ① DB에 암호화폐 거래소 가격 정보 Insert
- ② DB에서 암호화폐 가격 데이터 Import
- ③ 데이터 전처리 진행 (중복 데이터 처리, Feature 추가 등)
- ④ 전처리된 데이터로 모델링
- ⑤ 5분 마다 예측 시행
- ⑥ 공매도 or 공매수 포지션 예측 시 Telegram 메시지 전송
- ⑦ DB에 예측 신호 Insert
- ⑧ DB에 정답 레이블 Insert
- ⑨ 공매도 or 공매수 포지션 예측 시 Telegram 메시지로 적중 여부 전송
- ⑩ 매일 00:30(한국시간 기준)에 Telegram 메시지로일간 적중률 전송
- ⑪ 예측 신호를 바탕으로 푸시 알림 서비스 제공 및 자동 매매 진행

# 주요업무

## Modeling

- ❖ 목표 : n분 후 공매수 및 공매도 포지션 예측
  - ❖ 수익 얻기
  - ❖ 적중률 높이기
- ❖ Input : 암호화폐 가격 정보
- ❖ Output
  - ❖ 공매수 or
  - ❖ 공매도 or
  - ❖ 신호no

## ❖ Feature 연구

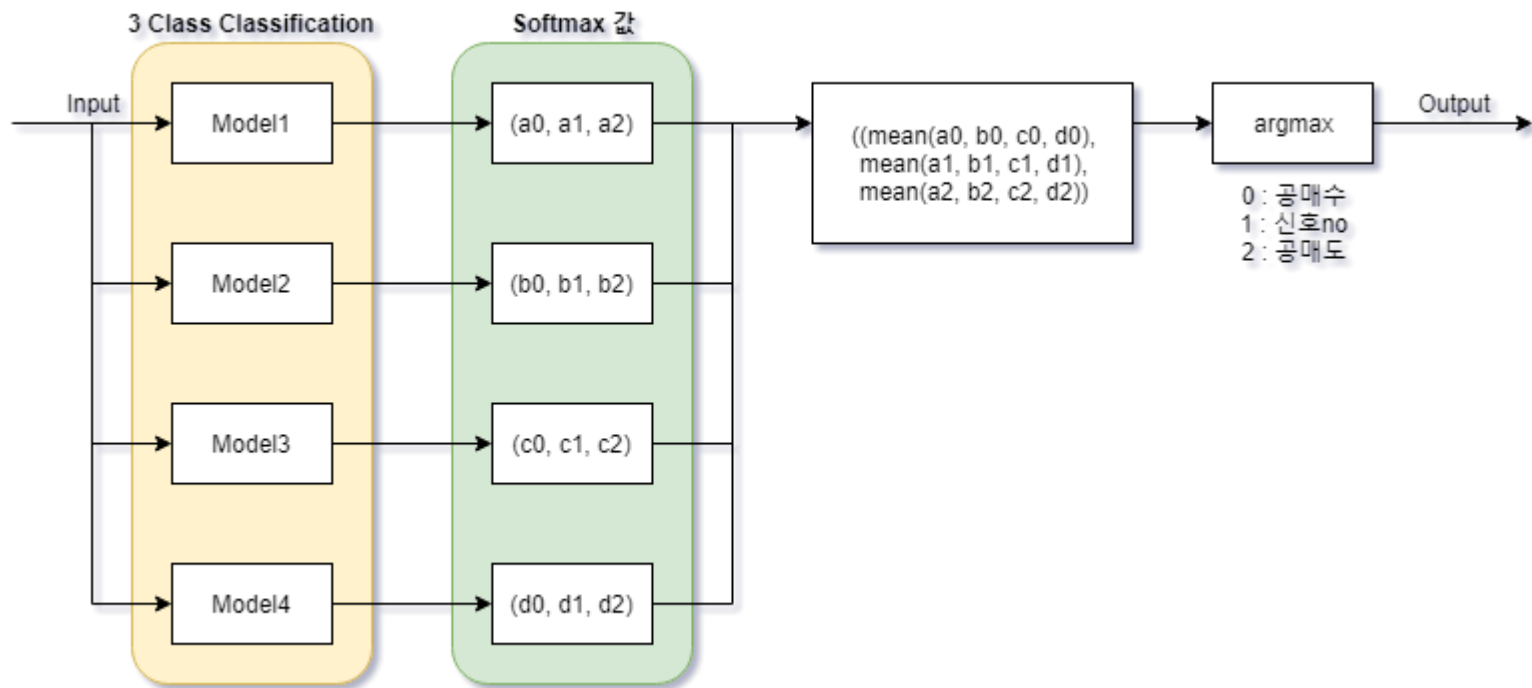
- ❖ 시계열 데이터 연구
- ❖ 투자 지표 연구

# 주요업무

## Modeling

### ❖ 1st Model

#### ❖ Model 구조



- ❖ Overfitting 해결 및 성능 개선을 위해 4개의 3 Class (공매수, 신호no, 공매도) Model 생성
- ❖ 각 모델의 Output인 Softmax 값들을 같은 인덱스끼리 평균 연산 후, 가장 큰 값을 가진 인덱스를 예측 값으로 전송
- ❖ 특징 및 문제점
  - ❖ 장대(큰 가격 변동)에 의존하여 예측하는 경향 강함 → 예측 실패의 경우 손실 큼
  - ❖ 가격 변동이 서서히 증가/감소하는 경우, 연달아 공매도/공매수 예측 → 적중률 감소 및 수수료 증가
  - ❖ 시장 변화에 따라 모델의 성능 편차가 매우 큼 → 자동 매매 서비스에 도입하기에는 불안정

## Modeling

### ❖ 2<sup>nd</sup> Model

#### ❖ Model 구조

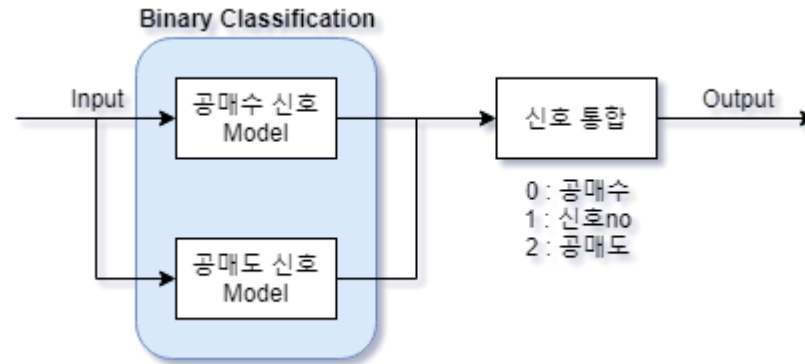
##### ❖ 2개의 Binary Model 생성

❖ n분 후 **공매수** 포지션인지 아닌지 예측하는 Model

❖ n분 후 **공매도** 포지션인지 아닌지 예측하는 Model

##### ❖ 두 모델의 Output을 하나의 신호로 통합

|        | 상승 No | 상승 Yes |
|--------|-------|--------|
| 하락 No  | 신호no  | 공매수    |
| 하락 Yes | 공매도   | 신호no   |



## Modeling

### ❖ 2<sup>nd</sup> Model

#### ❖ 목표 (개선 방향)

- ❖ 공매수와 공매도 class의 precision 증가를 목표로 모델링
- ❖ 연달아 신호(공매수 or 공매도)를 예측하지 않도록 모델링
- ❖ 장대(큰 가격 변동)에 민감하게 반응하지 않도록 모델링

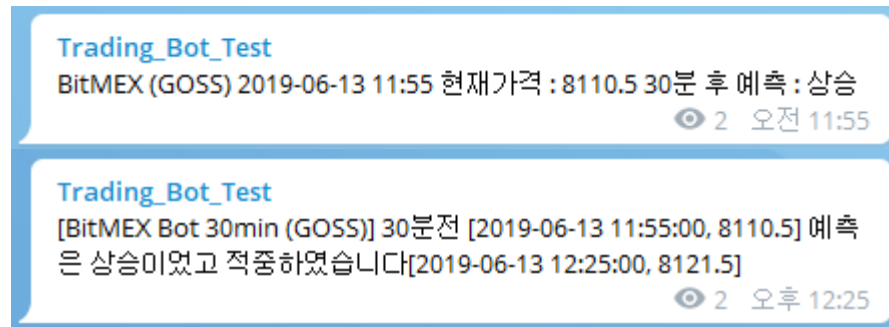
#### ❖ 개선 방안

- ❖ 클래스 불균형 문제 및 precision 개선을 위해 관련 Parameter Tuning
- ❖ Parameter Tuning을 통해 목적과 가장 적합한 값 지정
- ❖ 다양한 평가 지표 사용 (precision, 수익률, 신호 수 등)



## 프로세스 관리 및 유지보수

- ❖ 파이프라인의 ②~⑩ 프로세스 관리 및 모니터링
  - ❖ 적절한 시간에 프로세스가 작동하도록 Crontab 이용
  - ❖ Telegram을 통해 적절한 시간에 프로세스가 작동하는지 확인



- ❖ Git을 이용해 코드 관리
- ❖ 다수의 Model 관리

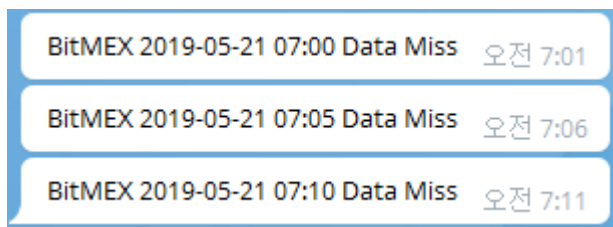
## 프로세스 관리 및 유지보수

### ❖ log 파일 관리

- ❖ Error 발생 시 원인을 파악하기 위해 일별로 log파일 생성
- ❖ Memory 관리를 위해 이상이 없으면 일주일 후 자동 삭제 (Shell Script 이용)

### ❖ 데이터 관리

- ❖ 암호화폐 가격 정보가 DB에 잘 쌓이는 지 모니터링
- ❖ DB에 데이터가 쌓이지 않을 경우, Telegram 메시지 전송 (실시간으로 확인)



## 주요 이슈 및 해결

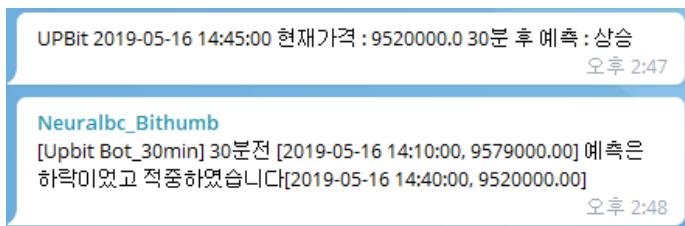
| 이슈  | 해결   |
|---|--|
| 모델에 많은 제약을 주기 위해 다양한 parameter의 tuning 필요 $\Rightarrow$ 모든 가지 수를 tuning 하기엔 많은 시간 소요                              | 각 parameter의 목적과 특징에 대해 공부하고, parameter 우선순위를 정해 tuning 진행 |
| 수익 및 적중률을 높이기 위해 적절한 보합 구간 조정 $\Rightarrow$ 보합 구간을 크게 하면 신호(공매수 or 공매도)를 예측하지 않고, 작게 하면 신호를 자주 예측 (클래스 불균형 문제 발생) | 보합 구간도 하나의 parameter로 간주하여 tuning 진행                       |
| 목적에 맞는 모델 선택  | 다양한 평가 지표를 사용하여 모델 선택                                      |

# 성과

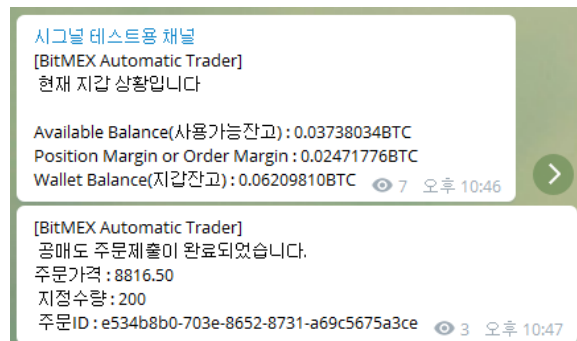
- ❖ 실시간 자동 매매 테스트 결과 최대 24% 수익 발생
- ❖ 유저들에게 푸시 알림 서비스 제공



Telegram



AiBitBip



AI Auto-Trading

