**SK네트웍스 Family AI 과정 12기  
 데이터 전처리 학습된 인공지능 모델**

|  |  |
| --- | --- |
| **산출물 단계** | 데이터 전처리 |
| **평가 산출물** | 학습된 인공지능 모델 |
| **제출 일자** |  |
| **깃허브 경로** | https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-2TEAM |
| **작성 팀원** | 윤 권 |

## 1. 모델 목적

#### 과거 주가 데이터를 기반으로 미래 가격의 변동 방향 혹은 수익률을 예측함으로써, 투자 판단을 지원하는 인공지능 예측 모델 구축 (강화학습 환경의 입력 또는 시그널 판단 근거로 활용)

## 2. 모델 아키텍처 설계

### 2.1 선정 모델

#### LSTM 기반 시계열 예측 모델

### 2.2 아키텍처 개요

#### 

|  |  |  |
| --- | --- | --- |
| 계층 | 구성 요소 | 역할 |
| 입력층 | OHLCV + 파생 변수 (5~7일) | 시계열 상태 벡터 입력 |
| LSTM | 128유닛 | 시계열 패턴 학습 및 요약 |
| Dropout | 비율 : 0.2 | 과적합 방지 |
| Dense | Fully Connected Layer | 수익률 또는 가격 예측값 출력 |

### 2.3 아키텍처 시각화

Input (5~7일 OHLCV+파생 데이터)

↓

LSTM Layer (1~2층, hidden size=64)

↓

Dropout (p=0.2)

↓

Dense Layer

↓

Output (예측 수익률 또는 가격)

### 2.4 설계 근거

#### LSTM은 시계열 데이터에서 시간적 흐름과 패턴을 학습하는 데 강점을 지님

#### 주가 예측에 필요한 과거 연속적 데이터를 효과적으로 반영할 수 있음

#### Dropout 및 Dense를 조합해 간단하지만 유연한 구조 구성

## 3. 모델 학습 요약

### 3.1 학습 데이터 구성

|  |  |
| --- | --- |
| **항목** | **수량 (건 단위)** |
| 학습 데이터 수 | 약 36,000 |
| 검증 데이터 수 | 약 9,000 |
| 평가(Test) 데이터 수 | 약 5,000 |

### 3.2 성능 평가 결과 (회귀 지표 기준)

|  |  |
| --- | --- |
| **지표** | **값** |
| MSE (평균제곱오차) | 0.0157 |
| MAE (평균절대오차) | 0.092 |
| RMSE (평균제곱근오차) | 0.125` |
| R^2 Score (결정계수) | 0.21 |
| 평균 예측 수익률 오차 (%) | ±4.38% |

### 3.3 해석 요약

#### LSTM 모델은 학습 안정성은 확보되었으니 예측 정확도는 낮은 편

#### 평균 오차±4.38%는 실거래 기준으로 활용하기엔 신뢰도 부족

#### R^2 값이 0.21로, 예측 모델로서의 설명력은 제한적

## 4. 저장 및 배포

### 4.1 저장 형식

|  |  |
| --- | --- |
| **항목** | **설명** |
| 저장 파일명 | best\_lstm\_model.pt |
| 저장 형식 | PyTorch .pt 파일 (state\_dict 방식) |
| 저장 방법 | torch.save(model.state\_dict(), ‘best\_lstm\_model’) |

### 4.2 GPU 최적화 설정

|  |  |
| --- | --- |
| **항목** | **적용 설정값** |
| 배치 크기 | 128 (RTX 5080의 대용량 메모리에 최적화) |
| LSTM 히든 크기 | 256 |
| DQN 히든 크기 | 512 (복합 모델 실험 시 사용) |
| 리플레이 버퍼 크기 | 50,000 |
| 혼합 정밀도 훈련 | 지원 (torch.cuda.amp 사용) |

### 4.3 모델 버전 관리 현황

#### 총 LSTM 모델 수: 12개

##### 각 모델은 다른 하이퍼파라미터 조합으로 실험

#### 최고 성능 모델: best\_lstm\_model.pt (RMSE 기준 최소)

#### 훈련 로그: 총 13개 (로그파일 활용하여 저장 관리 중)

### 4.4 모델 사양 요구 사항

|  |  |
| --- | --- |
| **항목** | **설명** |
| 프레임워크 | PyTorch 2.0 이상 |
| GPU | RTX 5080 GPU 이상 |
| 환경 설정 파일 | Requirements.txt 포함 |
| 혼합 정밀도 지원 | Torch.cuda.amp.autocast 활용하여 연산 효율성 향상 |

## 5. 종합 평가 및 활용 방안

### 5.1 모델 안정성

#### Best\_lstm\_model.pt는 여러 실험 모델 중 가장 낮은 RMSE를 기록한 모델로, 훈련 로그 및 하이퍼파라미터 기록과 함께 저장됨

#### 동일한 입력에 대해 재적재 시 ±0.1% 이내 오차 범위에서 일관된 추론 결과 확인

#### 모델 적재 및 추론 과정에 오류 없음, GPU 환경에서 실행 검증 완료

### 5.2 일반화 가능성

#### 테스트셋 기반 수익률 예측 정확도는 낮은 편이나, 다양한 종목에 대해 오버피팅 없이 일관된 수준의 성능 유지

#### 복잡한 패턴을 예측하기보다는 단순 변화 방향성 또는 보조 지표 용도로 일반화 가능성 있음

#### 시장 외부 변수(뉴스, 매크로 경제지표 등)가 포함되지 않아 예측 한계는 명확함

### 5.3 재사용성 및 응용 가능성

|  |  |
| --- | --- |
| **항목** | **설명** |
| 모델 저장 용량 | 약 1.2MB (state\_dict기준, 경량화 완료) |
| 추론 시간 | 평균 약 0.1초/건 (RTX 5080 기준) |
| 활용 형태 | 강화학습 환경 내 상태 입력으로 사용, 또는 방향성 분류 모델의 기반 특성 추출기로 사용 가능 |

### 5.4 향후 활용 방안

#### 현재 모델은 정확한 수익률 예측보다는 방향성 판단 또는 필터링 조건 생성 용도로 적합

#### 이후 모델 성능 개선을 위해 아래 방향으로의 활용 및 연구 계획 수립

|  |  |
| --- | --- |
| **계획** | **설명** |
| 입력 특성 확장 | 기술적 지표, 거래대금, 뉴스 감성 점수 등 외부 변수 추가 실험 |
| 출력 구조 수정 실험 | 수익률 회귀 → 상승/하락 분류 (Binary Classification)로 전환 실험 |
| DQN 연동 강화 | LSTM 예측값을 강화학습 에이전트의 정책 결정 보조 입력으로 활용 |
| 예측 신뢰도 기반 필터링 | 예측값 분포를 기반으로 높은 불확실성 데이터 제거 또는 알림 제공 |