📦 **요구사항 정의서 – indicators.py 모듈 업데이트 (최종 고급화 버전)**

**1. 프로젝트 개요**

* **모듈 이름**: indicators.py (통합 기술적 지표 계산 모듈)
* **목적**:
  + AI 자동화 트레이딩 프로그램에 사용될 고급 기술적 지표 계산 및 데이터 전처리 제공
  + 뉴스 데이터 및 감정 분석 통합
  + 실시간 데이터 처리 및 다중 시간 프레임 지원
  + AI 모델 학습 및 실시간 매매 신호 생성을 위한 데이터 전처리
  + 시장의 실시간 동향 및 복잡한 패턴을 분석하여 고급 트레이딩 전략 제공

**2. 주요 기능**

**2.1. 기존 기능:**

* **기존 지표** (9종):
  + **RSI (Relative Strength Index)**: 과매수/과매도 상태를 파악하여 가격 추세의 강도를 측정.
  + **MACD (Moving Average Convergence Divergence)**: 장단기 이동평균선의 차이를 통해 추세 전환 신호를 제공.
  + **VWAP (Volume Weighted Average Price)**: 거래량을 가중치로 한 평균 가격을 계산하여 공정한 시장 가격을 판단.
  + **Bollinger Bands**: 가격 변동성을 시각화하여 추세의 지속성 및 반전을 예측.
  + **ATR (Average True Range)**: 시장의 변동성을 측정하여 리스크 관리 및 손절매 설정에 사용.
  + **EMA (Exponential Moving Average)**: 최근 가격에 더 가중치를 두어 추세를 매끄럽게 표시.
  + **Fibonacci Retracement**: 중요한 지지 및 저항 수준을 예측하는 데 사용.
  + **Ichimoku Cloud**: 추세, 모멘텀, 지지 및 저항을 종합적으로 분석.
  + **Donchian Channel**: 고점 및 저점을 기반으로 추세를 판단.

**2.2. 추가될 기능 (정확히 20개 이상의 지표 구현):**

1. **트렌드 기반 지표 (3종 추가)**:
   * **Keltner Channel**: 가격의 평균과 ATR을 결합하여 추세 및 변동성을 분석.
   * **Supertrend**: 현재 가격과 ATR을 조합하여 강력한 추세 지표로 활용.
   * **Parabolic SAR**: 시간에 따른 추세의 방향과 포지션 전환 신호를 제공.
2. **모멘텀 기반 지표 (5종 추가)**:
   * **CCI (Commodity Channel Index)**: 가격의 이동평균과의 편차를 측정하여 과매수/과매도를 판단.
   * **ROC (Rate of Change)**: 가격의 상대적 변화율을 측정하여 추세 반전 또는 지속성을 평가.
   * **Williams %R**: 최고가 대비 현재 가격의 위치를 계산하여 과매수/과매도를 판단.
   * **Stochastic Oscillator**: 현재 가격과 최고/최저가의 관계를 분석하여 시장의 강도를 파악.
   * **ADX (Average Directional Index)**: 추세의 강도를 측정하며, 상승/하락 방향과는 무관.
3. **거래량 기반 지표 (3종 추가)**:
   * **MFI (Money Flow Index)**: 가격과 거래량을 결합하여 과매수/과매도 상태를 분석.
   * **Elder-Ray Index**: 매수와 매도 압력을 측정하여 시장의 힘을 평가.
   * **OBV (On-Balance Volume)**: 거래량의 흐름을 분석하여 시장의 추세를 파악.
4. **시장 복잡성 지표 (2종 추가)**:
   * **Hurst Exponent**: 가격 움직임의 랜덤성과 추세를 분석하여 시장의 복잡성을 평가.
   * **Fractal Dimension**: 시장의 패턴 복잡성을 측정하여 변동성과 추세를 분석.
5. **조정된 지표 (5종 추가)**:
   * **Adjusted RSI (감정 점수 결합)**: 뉴스 감정 점수를 반영하여 RSI 값을 조정.
   * **Sentiment-Adjusted MACD**: 감정 점수를 반영하여 MACD 신호의 신뢰도를 강화.
   * **Multi-Timeframe EMA**: 다양한 시간 프레임의 EMA를 결합하여 추세의 신뢰도를 강화.
   * **Adjusted VWAP**: 뉴스 데이터와 거래량 데이터를 반영한 가중 평균 가격.
   * **Sentiment-Weighted Bollinger Bands**: 뉴스 감정 데이터를 변동성 계산에 통합하여 정확도를 높임.

**총합: 기존 지표 9종 + 추가 지표 18종 = 총 27개의 지표**

**3. 주요 개선 사항**

**3.1. 지표 계산 로직 개선**

* **NumPy 및 Numba JIT 활용**: 반복 계산의 속도 향상.
* **지표 통합**: 단일 모듈에서 모든 지표를 계산하며, 중복 코드 제거.
* **벡터화된 처리**: 모든 계산이 벡터화된 방식으로 수행되어 성능을 극대화.
* **멀티스레딩 적용 (동시 계산 가능)**:
  + Python의 concurrent.futures를 활용하여 여러 지표를 병렬로 계산.
  + 각 스레드는 독립적인 데이터 세그먼트를 처리하며, 최종 결과를 통합.
  + 예시: RSI와 MACD를 동시에 계산하는 작업 분리.
* from concurrent.futures import ThreadPoolExecutor
* def calculate\_indicators(data):
* with ThreadPoolExecutor() as executor:
* executor.submit(calculate\_rsi, data)

executor.submit(calculate\_macd, data)

**3.2. 실시간 데이터 처리**

* WebSocket을 통해 실시간으로 데이터 수집.
* 실시간 시세 데이터를 기술적 지표 계산에 바로 활용.
* **스무딩 기법 적용 (노이즈 제거)**:
  + **Simple Moving Average (SMA)**: 데이터를 평활화하여 단기적 노이즈 제거.
  + **Gaussian Smoothing**: 더 부드럽고 연속적인 신호를 생성.

**3.3. 뉴스 데이터와의 통합**

* 뉴스 데이터를 기반으로 감정 점수를 계산하고, 이를 기술적 지표와 결합하여 조정된 지표를 생성.
* 감정 점수는 -1에서 1 사이로 스케일링.
* **코인니스 API 연동**:
  + https://api.coinness.com/v1/news API를 사용하여 뉴스 데이터를 수집.
  + 실시간 속보와 거래량 알림을 WebSocket으로 수신.
  + 중요 뉴스 키워드를 필터링하여 이벤트 발생 시 즉각적으로 지표에 반영.

**3.4. 다중 시간 프레임 분석**

* 여러 시간 단위에서 지표를 계산하여 통합 신호 생성.
* 다양한 시간 프레임 간 지표의 상관성을 활용하여 신호의 신뢰도 향상.

**3.5. AI 친화적 데이터 전처리 (정확성 향상)**

* **이상치에 강건한 스케일러 사용**:
  + RobustScaler: 이상치에 민감하지 않은 스케일링 방식으로 데이터 분포를 균등화.
  + Z-Score Scaling: 표준화 방식으로 데이터의 평균을 0, 표준편차를 1로 맞춤.
* **AI 모델 학습 최적화**:
* from sklearn.preprocessing import RobustScaler, StandardScaler
* def scale\_data(data):
* scaler = RobustScaler()

return scaler.fit\_transform(data)

**3.6. 보안 및 에러 처리**

* API Key는 환경 변수에서 로드하며, 코드에 직접 노출되지 않음.
* try-except 블록을 사용해 데이터 처리 중 발생할 수 있는 에러를 포괄적으로 관리.
* **로깅 시스템 강화**:
  + 거래량 급등/급락 시 알림을 기록.
  + 뉴스 데이터 감정 점수와 거래량 데이터를 통합한 이벤트 로그 작성.
  1. **GPU 가속 볼린저 밴드 계산**:
  + CUDA를 활용해 볼린저 밴드 계산의 속도를 대폭 향상.
  1. **Wavelet Transform 기반 노이즈 제거**:
  + 고급 Wavelet Transform을 사용해 데이터의 노이즈를 정밀하게 제거.
  1. **적응형 ATR 계산**:

시장 변동성에 따라 ATR 기간을 동적으로 조정하는 기능 추가.

**4. 데이터 흐름**

1. **데이터 입력**:
   * **실시간 시세 데이터**:
     + 업비트 WebSocket API를 통해 close, high, low, volume 데이터를 실시간으로 수집.
     + 데이터의 정확성을 보장하기 위해 업비트 REST API를 주기적으로 활용하여 WebSocket 데이터의 무결성을 검증.
   * **뉴스 데이터**:
     + 코인니스 API로 텍스트 기반 속보를 실시간 수집하며, 중요 키워드 필터링 적용.
     + 감정 분석을 수행하여 시장의 긍정/부정 점수를 도출.
   * **온체인 데이터 및 소셜 미디어 데이터**:
     + 주요 블록체인 지갑 활동 데이터 (Glassnode API).
     + 트위터 및 레딧의 소셜 트렌드 데이터를 API로 분석.
2. **기술적 지표 계산**:
   * Numba 및 NumPy로 지표 계산을 최적화.
   * **단계적 병렬 처리**:
     + concurrent.futures를 활용해 지표 계산 작업을 병렬로 처리.
     + 예: RSI, MACD, VWAP 계산을 독립적으로 수행하여 성능 향상.
   * 기존 및 신규 지표(27종)를 통합 계산하며, 지표별로 데이터프레임에 저장.
3. **뉴스 감정 분석 및 통합**:
   * **외부 API로 뉴스 데이터를 수집**:
     + TextBlob 라이브러리를 활용해 각 뉴스 항목의 감정 점수를 계산.
     + 코인니스 API와 WebSocket 기반 실시간 속보 데이터를 결합.
   * **감정 점수와 기술적 지표 결합**:
     + 감정 점수를 각 시점의 기술적 지표(RSI, MACD 등)에 가중치로 반영하여 조정된 지표 생성.
4. **다중 시간 프레임 데이터 생성**:
   * 1분, 5분, 1시간, 1일 단위로 데이터를 처리하여 시간 프레임별 지표를 생성.
   * 다중 시간 프레임 데이터 간 상관관계를 분석하여 종합 신호를 생성.
   * 추세 확인 및 이상치 제거를 위한 다중 시간 프레임 비교.
5. **결과 반환**:
   * 모든 계산 결과를 통합하여 단일 DataFrame으로 정리.
   * DataFrame에는 각 시간 프레임 및 감정 점수 통합 결과가 포함됨.
   * 결과는 AI 모델로 입력되며, 매매 신호 생성 및 전략 테스트에 활용.

**5. 디렉터리 구조**

모듈은 단일 파일로 유지하며, 모든 기능을 통합:

indicators.py # 모든 지표 계산, 데이터 처리 및 통합 기능 포함

**6. 요구 기술 스택**

* **언어**: Python
* **필수 라이브러리**:
  + Numpy, Pandas, Numba: 데이터 처리 및 최적화.
  + TextBlob: 뉴스 감정 분석.
  + Requests: 외부 API 통합.
  + WebSocket: 실시간 데이터 수집.
  + Scikit-learn: 데이터 스케일링 및 전처리.
* **환경 변수 관리**: python-dotenv를 활용한 .env 파일.
* **로깅**: logging 모듈로 시스템 이벤트 기록.