\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ANALİZ TÜRLERİ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

| **Analiz Türü** | **Kullanım Alanı** | **Kimler İçin Uygun?** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| **Teknik Analiz** | Grafikler ve indikatörler | Kısa vadeli traderlar |

|  |  |  |
| --- | --- | --- |
| **Temel Analiz** | Şirket finansalları, haberler | Uzun vadeli yatırımcılar |

|  |  |  |
| --- | --- | --- |
| **SWOT Analizi** | Şirketlerin güçlü/zayıf yönleri | Hisse yatırımcıları |

|  |  |  |
| --- | --- | --- |
| **Sentiment Analizi** | Sosyal medya ve piyasa psikolojisi | Kripto traderları |

|  |  |  |
| --- | --- | --- |
| **Zincir Üstü (On-Chain) Analiz** | Blok zincir verileri | Kripto yatırımcıları |

|  |  |  |
| --- | --- | --- |
| **Piyasa Döngü Analizi** | Ekonomik döngüler | Uzun vadeli yatırımcılar |

|  |  |  |
| --- | --- | --- |
| **Elliot Dalga Teorisi** | Dalga formasyonları | Teknik analiz yapanlar |

|  |  |  |
| --- | --- | --- |
| **Arbitraj Analizi** | Fiyat farklarıyla kazanç | Hızlı işlem yapanlar |

|  |  |  |
| --- | --- | --- |
| **Takvimsel Analiz** | Mevsimsel fiyat hareketleri | Sezonluk yatırımcılar |

|  |  |  |
| --- | --- | --- |
| **Kuantum Finans ve Stokastik Analiz** | İleri düzey matematik | Profesyonel yatırımcılar |

|  |  |  |
| --- | --- | --- |
| **Makine Öğrenmesi ve AI Analiz** | Büyük veri ve tahminler | Algoritmik traderlar |

|  |  |  |
| --- | --- | --- |
| **Hacim ve Likidite Analizi** | Piyasa derinliği ve emir defteri | Scalper ve gün içi traderlar |

|  |  |  |
| --- | --- | --- |
| **Risk Analizi ve Portföy Optimizasyonu** | Portföy yönetimi | Kurumsal yatırımcılar |

|  |  |  |
| --- | --- | --- |
| **Opsiyon ve Vadeli İşlemler Analizi** | Vadeli işlemler piyasası | Opsiyon yatırımcıları |

|  |  |  |
| --- | --- | --- |
| **Şirket İçgörü Analizi** | Hissedar hareketleri | Hisse senedi yatırımcıları |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SCALPİNG İÇİN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evet, **scalping ve gün içi alım-satım için en etkili analiz yöntemleri bunlar**. Ancak, **stratejinin başarısı piyasaya, varlık türüne ve volatiliteye bağlı olarak değişebilir**. Eğer daha da geliştirmek istersen, şu ek analizleri de dahil edebiliriz:

**📌 Sonuç: Kullanılacak Analizler**

| **Analiz Türü** | **Kullanım Amacı** | **Gerekli mi?** |
| --- | --- | --- |
| **Teknik Analiz** | Giriş-çıkış noktalarını belirleme | ✅ Evet (Temel şart) |
| **Hacim Analizi** | Büyük yatırımcı hareketlerini izleme | ✅ Evet (Scalping için kritik) |
| **Sentiment Analizi** | Piyasa psikolojisini ölçme | ✅ Evet (Hızlı hareket eden piyasalarda etkili) |
| **Yapay Zeka & ML** | Strateji optimizasyonu ve tahminleme | ✅ Evet (AI tabanlı karar mekanizması için) |
| **Piyasa Döngü Analizi** | En iyi işlem saatlerini belirleme | 🔶 Opsiyonel (Avantaj sağlar) |
| **Takvim & Haber Analizi** | Volatilite yaratabilecek haberlerden kaçınma | 🔶 Opsiyonel (Risk yönetimi için iyi olur) |
| **Market Making** | Spread ve likidite fırsatlarını değerlendirme | 🔶 Opsiyonel (Gelişmiş bir strateji için) |
| **Manipülasyon Tespiti** | Büyük oyuncuların suni hareketlerini anlama | 🔶 Opsiyonel (Özellikle kriptoda faydalı) |

✅ **Şu an belirlediğimiz 4 ana analiz türü kesinlikle yeterli.**  
🔶 **Ek analizler ise botu daha akıllı ve güvenli hale getirebilir.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Teknik Analiz |  |  |  |  |  | ✅ Evet | Giriş/çıkış noktaları ve momentum indikatörleri (RSI, MACD). |
| Hacim Analizi |  |  |  |  |  | ✅ Evet | Büyük emir bloklarını ve anormal hacim artışlarını izleme. |
| Risk Yönetimi |  |  |  |  |  | ✅ Evet | Stop-loss, take-profit, lot boyutu optimizasyonu. |
| Latans Yönetimi |  |  |  |  |  | ✅ Evet | Düşük gecikmeli veri akışı ve emir iletimi. |
| Sentiment Analizi |  |  |  |  |  | 🔶 Opsiyonel | Sosyal medya ve haberlerden anlık veri çekme. |
| Yapay Zeka & ML |  |  |  |  |  | 🔶 Opsiyonel | Dinamik strateji optimizasyonu ve anomali tespiti. |
| Mikro Piyasa Yapısı |  |  |  |  |  | 🔶 Opsiyonel | Emir defteri derinliği ve spread dalgalanmalarını izleme. |
| İşlem Maliyetleri |  |  |  |  |  | ✅ Evet | Komisyon, spread ve slippage hesaplamaları. |

Onchain , cvd analizleri, likiditasyonlar, yapay zeka doğruluk oranı , bütçeni yazacaksın sana para yönetimini de söyleyecek. Formasyonlar, risk yönetimi para , **🔹 Kripto Paralar:**

* **ABD ve Avrupa piyasalarının açıldığı saatlerde işlem hacmi artar.**
* **08:00 - 12:00 ve 19:00 - 23:00 en iyi saatlerdir.**
*  **25+ Formasyon Desteği:**
*  Tüm trend dönüş ve devam formasyonları
*  Üçgen, kama, bayrak çeşitleri
*  Hacim analizli bayrak/flama tespiti
*  **Akıllı Parametre Yönetimi:**
* 
* params = {
* 'formasyon\_fark\_limit': 0.03,  # %3 fark toleransı
* 'ucgen\_egim\_limit': 0.005,      # Üçgen eğim hassasiyeti
* 'hacim\_artis\_oran': 1.2         # Bayrak/flama hacim artışı
* }
*  **Güvenilirlik Skorlama:**
*  Formasyon tipine göre otomatik skorlama
*  Tarihsel verilere dayalı risk analizi4. **Detaylı Raporlama:**
*  Zaman damgası ve fiyat bilgisi
*  Formasyonların kategorizasyonu
*  Olası trend dönüş/devam uyarıları

Evet, Python ile \*\*Binance API\*\*'sini kullanarak gerçek zamanlı verileri çekip, bu verileri \*\*grafiklere dökebilir\*\*, hatta \*\*al-sat sinyallerinizi test edebilirsiniz\*\*. İşte adım adım çözüm ve cevaplar:

---

### 1. \*\*Veri Çekmek Zahmetli mi?\*\*

- \*\*Hayır\*\*, Binance API'nin kapsamlı dokümantasyonu ve Python kütüphaneleri (örneğin, `python-binance`, `ccxt`) sayesinde veri çekmek oldukça kolay.

- \*\*Örnek Kod (Tarihsel Veri):\*\*

```python

from binance import Client

client = Client(api\_key, api\_secret)

historical\_data = client.get\_historical\_klines(

symbol="BTCUSDT",

interval=Client.KLINE\_INTERVAL\_1HOUR,

start\_str="1 week ago UTC"

)

```

- \*\*WebSocket ile Gerçek Zamanlı Veri:\*\*

```python

from binance import ThreadedWebsocketManager

def handle\_socket\_message(msg):

print(msg)

twm = ThreadedWebsocketManager(api\_key, api\_secret)

twm.start()

twm.start\_kline\_socket(callback=handle\_socket\_message, symbol="BTCUSDT")

```

---

### 2. \*\*Grafikleri Nasıl Görselleştirebilirim?\*\*

- \*\*Matplotlib/Plotly\*\* ile statik veya interaktif grafikler oluşturabilirsiniz.

- \*\*Örnek (Matplotlib):\*\*

```python

import pandas as pd

import matplotlib.pyplot as plt

df = pd.DataFrame(historical\_data, columns=['timestamp', 'open', 'high', 'low', 'close', 'volume', 'close\_time', 'quote\_asset\_volume', 'number\_of\_trades', 'taker\_buy\_base', 'taker\_buy\_quote', 'ignore'])

df['close'] = df['close'].astype(float)

df['timestamp'] = pd.to\_datetime(df['timestamp'], unit='ms')

plt.figure(figsize=(12,6))

plt.plot(df['timestamp'], df['close'])

plt.title("BTC/USDT Fiyat Grafiği")

plt.show()

```

- \*\*Daha Profesyonel Görseller İçin:\*\*

- \*\*Plotly\*\* veya \*\*mplfinance\*\* kullanın (örneğin, mum grafikleri ve teknik göstergeler ekleyebilirsiniz).

- \*\*TradingView Benzeri Arayüz:\*\* `plotly.graph\_objects` ile interaktif grafikler yapabilirsiniz.

---

### 3. \*\*Backtest ve Canlı Test Yapabilir miyim?\*\*

- \*\*Evet!\*\* Tarihsel verilerle \*\*backtest\*\* yapabilir, canlı verilerle de stratejinizi test edebilirsiniz.

- \*\*Backtest için Kütüphaneler:\*\*

- \*\*Backtrader\*\*, \*\*vectorbt\*\*, veya \*\*PyAlgoTrade\*\* gibi kütüphanelerle otomatize edebilirsiniz.

- \*\*Örnek (Backtrader):\*\*

```python

import backtrader as bt

class MyStrategy(bt.Strategy):

def \_\_init\_\_(self):

self.sma = bt.indicators.SMA(period=20)

def next(self):

if self.data.close[0] > self.sma[0]:

self.buy()

elif self.data.close[0] < self.sma[0]:

self.sell()

cerebro = bt.Cerebro()

cerebro.addstrategy(MyStrategy)

data = bt.feeds.PandasData(dataname=df)

cerebro.adddata(data)

cerebro.run()

cerebro.plot()

```

- \*\*Canlı Test:\*\*

- \*\*Paper Trading (Demo Hesap):\*\* Binance Futures testnet kullanarak gerçek para riski olmadan test yapabilirsiniz.

- \*\*WebSocket\*\* ile canlı fiyatları alıp, stratejinizi anlık çalıştırabilirsiniz.

---

### 4. \*\*Ücretsiz mi?\*\*

- \*\*Evet!\*\*

- Binance API ücretsizdir (belirli bir request limiti dahilinde).

- Python kütüphaneleri (`python-binance`, `backtrader`, `plotly`) açık kaynaklı ve ücretsizdir.

- \*\*Not:\*\* Yüksek frekanslı veri çekiyorsanız veya çok fazla istek atıyorsanız, Binance API limitlerine dikkat edin.

---

### 5. \*\*Zorlukları ve Tavsiyeler\*\*

- \*\*Zorluklar:\*\*

- WebSocket bağlantılarını yönetmek başlangıçta karmaşık gelebilir.

- Gerçek zamanlı veri ile backtest arasında \*\*sürüklenme (slippage)\*\* farklılıkları olabilir.

- \*\*Tavsiyeler:\*\*

1. Önce \*\*tarihsel verilerle backtest\*\* yapın.

2. Stratejinizi \*\*Demo Hesap\*\* üzerinde test edin.

3. Grafikleri görselleştirmek için \*\*Jupyter Notebook\*\* kullanın (hızlı iterasyon için ideal).

4. Kodunuzu modüler hale getirin (veri çekme, sinyal üretme, backtest ayrı dosyalarda olsun).

---

### 6. \*\*Örnek Çalışma Akışı\*\*

1. \*\*Veri Çekme:\*\* `python-binance` ile BTC/USDT 1 saatlik verileri alın.

2. \*\*Sinyal Üretme:\*\* Algoritmanızı yazın (örneğin, MACD kesişimleri).

3. \*\*Görselleştirme:\*\* `mplfinance` ile mum grafiklerine sinyalleri işaretleyin.

4. \*\*Backtest:\*\* `vectorbt` ile stratejinizin geçmiş performansını ölçün.

5. \*\*Canlı Test:\*\* WebSocket ile gerçek zamanlı veri alıp, demo hesapta işlem emri gönderin.

---

### Sonuç

\*\*Evet, yapabilirsiniz!\*\* Python ile hem grafikleri Binance'taki gibi görselleştirebilir, hem de ücretsiz araçlarla backtest ve canlı test yapabilirsiniz. Başlangıçta zaman alabilir, ancak doğru kütüphaneleri kullanarak süreci kolaylaştırabilirsiniz. İlk adım olarak \*\*Jupyter Notebook\*\* üzerinde basit bir mum grafiği + SMA stratejisi oluşturup test edebilirsiniz.

### ****Backtest Motoru:****vectorbt****veya****backtrader

ssh -i "$env:USERPROFILE\.ssh\yeni-anahtar.pem" [ubuntu@13.51.172.34](mailto:ubuntu@13.51.172.34)

 Grafik tasarımını geliştir

 Hata yönetimini ekle

 Logging mekanizması ekle

 Güvenlik önlemlerini artır

https://13.51.172.34.nip.io