

Value Transfer Predictor

Finansal Varlıklar Arası Değer Transfer Analizi ve Tahmin Sistemi

Bu proje, finansal varlıklar (Bitcoin, altın, gümüş, petrol, hisse senetleri vb.) arasındaki değer transfer ilişkilerini analiz eden ve gelecekteki değerleri tahmin eden bir machine learning sistemidir.

Proje Amacı

Finansal piyasalarda "bir değer yükselirken, başka bir değer düşer" teorisini test etmek ve bu ilişkileri kullanarak gelecekteki fiyat hareketlerini tahmin etmek.

Özellikler

Veri Analizi

- Korelasyon Analizi:** Varlıklar arası ilişki gücünü hesaplama
- Değer Transfer Analizi:** Hangi varlıkların ters yönde hareket ettiğini belirleme
- İstatistiksel İnceleme:** Aykırı değer tespiti ve veri temizleme

Machine Learning Modelleri

- Linear Regression:** Doğrusal ilişki tabanlı tahminleme
- Random Forest:** Ensemble learning ile güçlü tahminler
- Feature Engineering:** Oransal ilişkiler, moving averages, volatilite göstergeleri

Web Arayüzü

- Interactive Dashboard:** Gerçek zamanlı tahmin ve analiz
- Görselleştirme:** Plotly ile interaktif grafikler
- Karşılaştırma:** İki varlık arasında detaylı analiz

Kurulum

Pip ile Kurulum

```
bash
```

```
pip install value-transfer-predictor
```

Manuel Kurulum

bash

```
git clone https://github.com/yourusername/value-transfer-predictor.git
cd value-transfer-predictor
pip install -e .
```

Gereksinimler

bash

```
pip install -r requirements.txt
```

Dosya Yapısı

```
value-transfer-predictor/
├── src/
│   └── value_transfer_predictor/
│       ├── __init__.py
│       ├── analyzer.py ..... # Ana analiz sınıfı
│       ├── api.py ..... # Flask API
│       ├── models.py ..... # ML modelleri
│       └── cli.py ..... # Komut satırı arayüzü
├── tests/
│   ├── test_analyzer.py
│   ├── test_models.py
│   └── sample_data.csv
└── templates/
    └── index.html ..... # Web arayüzü
└── requirements.txt
└── pyproject.toml
└── README.md
└── LICENSE
```



Kullanım

1. Python Kodu ile Kullanım

```
python

from value_transfer_predictor import ValueTransferPredictor
import pandas as pd

# Veri yükleme
data = pd.read_csv('financial_data.csv')
predictor = ValueTransferPredictor()

# Veriyi işleme
predictor.load_data(data)

# Korelasyon analizi
correlations = predictor.analyze_correlations()
print(correlations)

# Değer transfer analizi
transfer_patterns = predictor.analyze_value_transfer()
print(transfer_patterns)

# Model eğitimi
predictor.train_models()

# Tahmin yapma
predictions = predictor.predict(['BTC', 'GOLD'], days=30)
print(predictions)
```

2. Komut Satırı Kullanımı

```
bash

# Temel analiz
vtp analyze --data financial_data.csv --assets BTC,GOLD,OIL

# Tahmin yapma
vtp predict --data financial_data.csv --target BTC --days 30

# Web arayüzüni başlatma
vtp serve --port 5000
```

3. Web Arayüzü

```
bash
```

```
# Sunucuyu başlat
python -m value_transfer_predictor.api

# Tarayıcıda açın
# http://localhost:5000
```

API Kullanımı

Endpoint'ler

```
bash
```

```
# Analiz endpoint'i
POST /api/analyze
{
    "assets": ["BTC", "GOLD", "OIL"],
    "start_date": "2023-01-01",
    "end_date": "2024-01-01"
}

# Tahmin endpoint'i
POST /api/predict
{
    "target_asset": "BTC",
    "reference_assets": ["GOLD", "OIL"],
    "days": 30
}

# Korelasyon endpoint'i
GET /api/correlations?assets=BTC,GOLD,OIL
```

Örnek API Kullanımı

```
python

import requests

# Analiz isteği
response = requests.post('http://localhost:5000/api/analyze', json={
    'assets': ['BTC', 'GOLD', 'OIL'],
    'start_date': '2023-01-01',
    'end_date': '2024-01-01'
})

analysis_result = response.json()
print(analysis_result)
```

🔧 Konfigürasyon

config.yaml

```
yaml

data:
  source: "yfinance" # yfinance, alpha_vantage, Local_csv
  update_interval: 3600 # saniye

models:
  default: "random_forest"
  hyperparameters:
    random_forest:
      n_estimators: 100
      max_depth: 10
    linear_regression:
      fit_intercept: true

api:
  host: "0.0.0.0"
  port: 5000
  debug: false

visualization:
  theme: "plotly_dark"
  default_period: 365
```

📈 Desteklenen Varlıklar

Kripto Paralar

- Bitcoin (BTC)

- Ethereum (ETH)
- Litecoin (LTC)
- Ripple (XRP)

Emtialar

- Altın (GOLD)
- Gümüş (SILVER)
- Petrol (OIL)
- Doğalgaz (NATGAS)

Hisse Senetleri

- S&P 500 (SPY)
- NASDAQ (QQQ)
- Apple (AAPL)
- Tesla (TSLA)

Döviz Kurları

- USD/EUR
- USD/TRY
- USD/GBP
- USD/JPY

Test Etme

```
bash

# Tüm testleri çalıştır
pytest tests/

# Belirli test dosyası
pytest tests/test_analyzer.py

# Kapsamlı test raporu
pytest tests/ --cov=value_transfer_predictor --cov-report=html
```

Performans Metrikleri

Model Değerlendirme

- **RMSE** (Root Mean Square Error)

- **MAE** (Mean Absolute Error)
- **R²** (Coefficient of Determination)
- **MAPE** (Mean Absolute Percentage Error)

Örnek Sonuçlar

Model Performance Results:

Model	RMSE	MAE	R ²
Random Forest	0.0234	0.0187	0.8456
Linear Reg.	0.0312	0.0245	0.7823
XGBoost	0.0198	0.0156	0.8734

⌚ Gerçek Zamanlı Veri

Veri Kaynakları

```
python

# Yahoo Finance entegrasyonu
from value_transfer_predictor.data_sources import YahooFinanceSource

source = YahooFinanceSource()
data = source.get_real_time_data(['BTC-USD', 'GC=F', 'CL=F'])

# Alpha Vantage entegrasyonu
from value_transfer_predictor.data_sources import AlphaVantageSource

source = AlphaVantageSource(api_key='your_api_key')
data = source.get_historical_data('BTC', period='1y')
```

📱 Mobil Destek

Web arayüzü responsive tasarım ile mobil cihazlarda da kullanılabilir.

PWA Desteği

- Offline çalışma
- Ana ekran ekleme
- Push bildirimleri

🔧 Geliştirici Rehberi

Yeni Model Ekleme

python

```
from value_transfer_predictor.models import BaseModel

class YourCustomModel(BaseModel):
    def __init__(self):
        super().__init__()
        self.model = YourMLModel()

    def train(self, X, y):
        self.model.fit(X, y)

    def predict(self, X):
        return self.model.predict(X)
```

Yeni Veri Kaynağı Ekleme

python

```
from value_transfer_predictor.data_sources import BaseDataSource

class YourDataSource(BaseDataSource):
    def get_data(self, symbol, period):
        # Veri alma implementasyonu
        return data
```

Hata Ayıklama

Yaygın Problemler

- Veri Kaynağı Hatası:** API key kontrolü yapın
- Model Eğitimi Hatası:** Veri formatını kontrol edin
- Tahmin Hatası:** Eksik feature'ları kontrol edin

Debug Modu

python

```
import logging
logging.basicConfig(level=logging.DEBUG)

predictor = ValueTransferPredictor(debug=True)
```

Lisans

MIT License - Detaylar için [LICENSE](#) dosyasına bakın.

Katkıda Bulunma

1. Fork yapın
2. Feature branch oluşturun (`git checkout -b feature/AmazingFeature`)
3. Commit yapın (`git commit -m 'Add some AmazingFeature'`)
4. Branch'i push edin (`git push origin feature/AmazingFeature`)
5. Pull Request açın

İletişim

- **Email:** your.email@example.com
- **GitHub:** [@yourusername](https://github.com/@yourusername)
- **LinkedIn:** [Your Name](https://www.linkedin.com/in/Your Name)

Teşekkürler

Bu projeyi geliştirmede yardımcı olan:

- Yahoo Finance API
- Scikit-learn Community
- Plotly Development Team
- Open Source Community

Referanslar

1. "Modern Portfolio Theory" - Harry Markowitz
2. "Machine Learning for Asset Management" - Marcos López de Prado
3. "Algorithmic Trading" - Ernest P. Chan
4. Python for Finance - Yves Hilpisch

 Projeyi beğendiyseniz star vermeyi unutmayın!