

# Részvények becslése lineáris regresszióval

W93NMV

(<https://colab.research.google.com/drive/1Ax21KfRVWncJrcKIKvjWGmuj9hqck8E3?usp=sharing>)

## Feladat

A beadandó feladat célja a Bitcoin (BTC-USD) és Ethereum (ETH-USD) tőzsdei részvények közötti összefüggés vizsgálata és néhány elemzés végrehajtása. Az elemzések magukban foglalják a lineáris regressziót melynek segítségével előrejelezzük az árfolyamot, páros kereskedést és egy mozgóátlag kiszámítását.

### *Előkészület (adatok letöltése, importálás)*

```
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression

ticker1 = "BTC-USD"
ticker2 = "ETH-USD"

# A start megadása nem fontos

df = yf.download(f"{ticker1},{ticker2}", start="2018-01-01")
```

## Lineáris regresszió

```
# Százalékosan meghatározzuk a zárási árfolyamot (ha nem kriptovaluta
lenne akkor az Adj Close pontosabb értéked ad mint a Close)

df[ticker1 + '_change'] = df['Adj Close'][ticker1].pct_change() + 1
df[ticker2 + '_change'] = df['Adj Close'][ticker2].pct_change() + 1

# Üres sorokat kidobjuk (ha nem kriptovaluta lenne hanem például
amerikai tech részvény akkor a szünnapok miatt üres lenne)
df = df.dropna()

# Előkészítés a modellre

X = df[ticker2+'_change'].values.reshape(-1, 1)
y = df[ticker1+'_change'].values

model = LinearRegression()
model.fit(X, y)

# Korrelációs együttható, regresszió egyenes, R^2 kiszámítása
slope = model.coef_[0]
intercept = model.intercept_

correlation_coefficient = np.corrcoef(X.T, y.T)[0, 1]
r_squared = correlation_coefficient**2

print(f"Korrelációs együttható: {correlation_coefficient:.4f}")
print(f"R-négyzet érték: {r_squared:.4f}")
print(f'Regresszió egyenes: {ticker1} = {slope} * {ticker2} +
{intercept}')

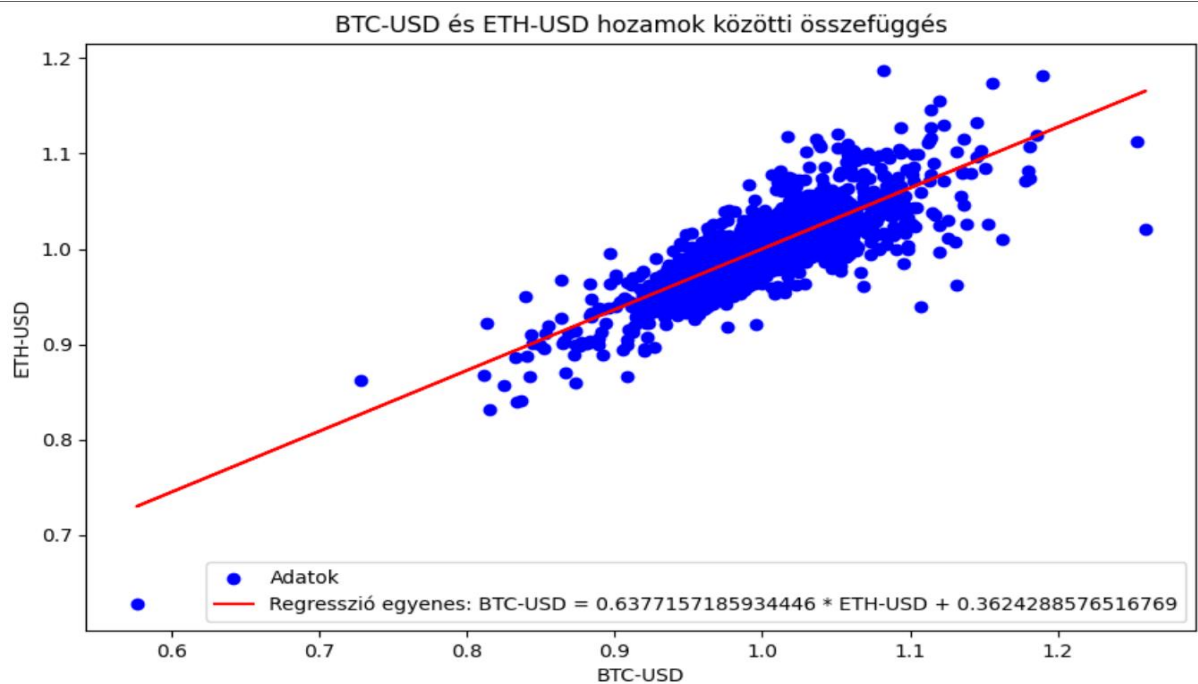
# Kirajzolás
plt.figure(figsize=(10, 6))
plt.scatter(X, y, label='Adatok', color='blue')
plt.plot(X, model.predict(X), color='red', label=f'Regresszió egyenes:
{ticker1} = {slope} * {ticker2} + {intercept}')
plt.xlabel(ticker1)
plt.ylabel(ticker2)
plt.legend()
plt.title(f'{ticker1} és {ticker2} hozamok közötti összefüggés')
plt.show()
```

Azt feltételezem hogy ha a BTC értéke növekszik akkor az ETH-nak is növekednie kell, ugyan ez igaz fordítva.

1. Százalékosan meghatározzuk a zárási árfolyamot (ha nem kriptovaluta lenne akkor az Adj Close pontosabb érték ad mint a Close)
2. Üres sorokat kidobjuk (ha nem kriptovaluta lenne hanem például részvény akkor a munkaszüneti napok miatt üres lenne)
3. Modellt előkészítése
4. Korrelációs együttható, regresszió egyenes,  $R^2$  kiszámítása
  - a. Korrelációs együttható megadja mennyire hasonlóan mozognak, ebben az esetben az eredmény 0.8202 lett ami azt jelenti, hogy erősen együtt mozognak.
  - b. Regresszió egyenes modellezi hogyan változik az egyik (BTC) - (ETH) értékének függvényében.

$BTC-USD = 0.63771571859 * ETH-USD + 0.36242885765$  Pld.: BTC árfolyama magasabb amikor az ETH nulla.

- c.  $R^2$  eredménye a modell illeszkedésének minősége, értékem 0.6860 azt jelenti, hogy viszonylag jó értékben magyarázza meg a célváltozót a független változók segítségével, itt is lehet hiba ami miatt torzít.



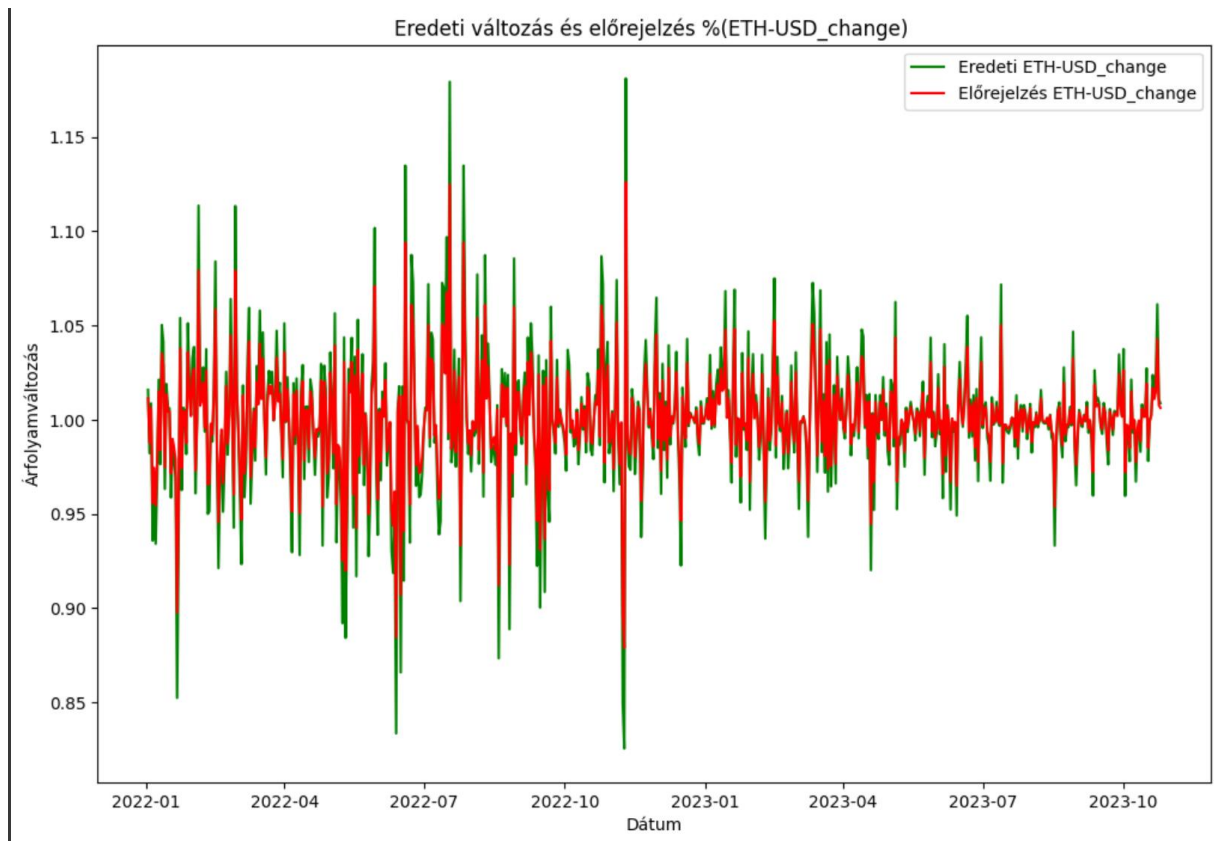
## Eredeti és előre jelzett százalékos változás összehasonlítása (2022-)

```
df['predict_LR'] = model.predict(X)

plt.figure(figsize=(12, 8))
plt.plot(df.index, df[ticker2+'_change'], label=f'Eredeti {ticker2}_change', color='green')

plt.plot(df.index, df['predict_LR'], label=f'Előrejelzés {ticker2}_change', color='red')

plt.xlabel('Dátum')
plt.ylabel('Árfolyamváltozás')
plt.legend()
plt.title(f'Eredeti változás és előrejelzés %({ticker2}_change) ')
plt.show()
```



Látható, hogy tévedett a modell az előrejelzést illetően hiszen nem csak a másik valuta befolyásolja az árfolyamot hanem többi más tényező is, 2022-től vizsgáltam az átláthatóság miatt

## Árfolyam becslése

Ugyan az, mint amikor a százalékos előrejelzést csináltam, csak itt a zárási értékre vagyok kíváncsi és ez alapján nem százalékosan jelezzük előre hanem mint értékkülönbség.

```
# Nem a százalékos zárást hanem érték szerintit nézek

df[ticker1 + '_change'] = df['Adj Close'][ticker1].diff()
df[ticker2 + '_change'] = df['Adj Close'][ticker2].diff()

df = df.dropna()

# Modell illesztés
X = df[ticker2+ '_change'].values.reshape(-1, 1)
y = df[ticker1+ '_change'].values

model = LinearRegression()
model.fit(X, y)
df['predict_LR'] = model.predict(X)
```

```

# Rajzolás, egyéb infó
min_value = df[ticker2+'_change'].min()
max_value = df[ticker2+'_change'].max()

min_date = df[ticker2+'_change'].idxmin()
max_date = df[ticker2+'_change'].idxmax()

plt.figure(figsize=(12, 6))
plt.plot(df.index, df[ticker2+'_change'], label=f'Eredeti
{ticker2}_change', color='green')
plt.plot(df.index, df['predict_LR'], label=f'Előrejelzés
{ticker2}_change', color='red')

plt.scatter(min_date, min_value, marker='o', color='brown',
label=f'Min: {min_value:.2f} ({min_date})')
plt.scatter(max_date, max_value, marker='o', color='black',
label=f'Max: {max_value:.2f} ({max_date})')

plt.xlabel('Dátum')
plt.ylabel('Árfolyamváltozás')
plt.legend()
plt.title(f'Eredeti változás és előrejelzés ({ticker2}_change)')

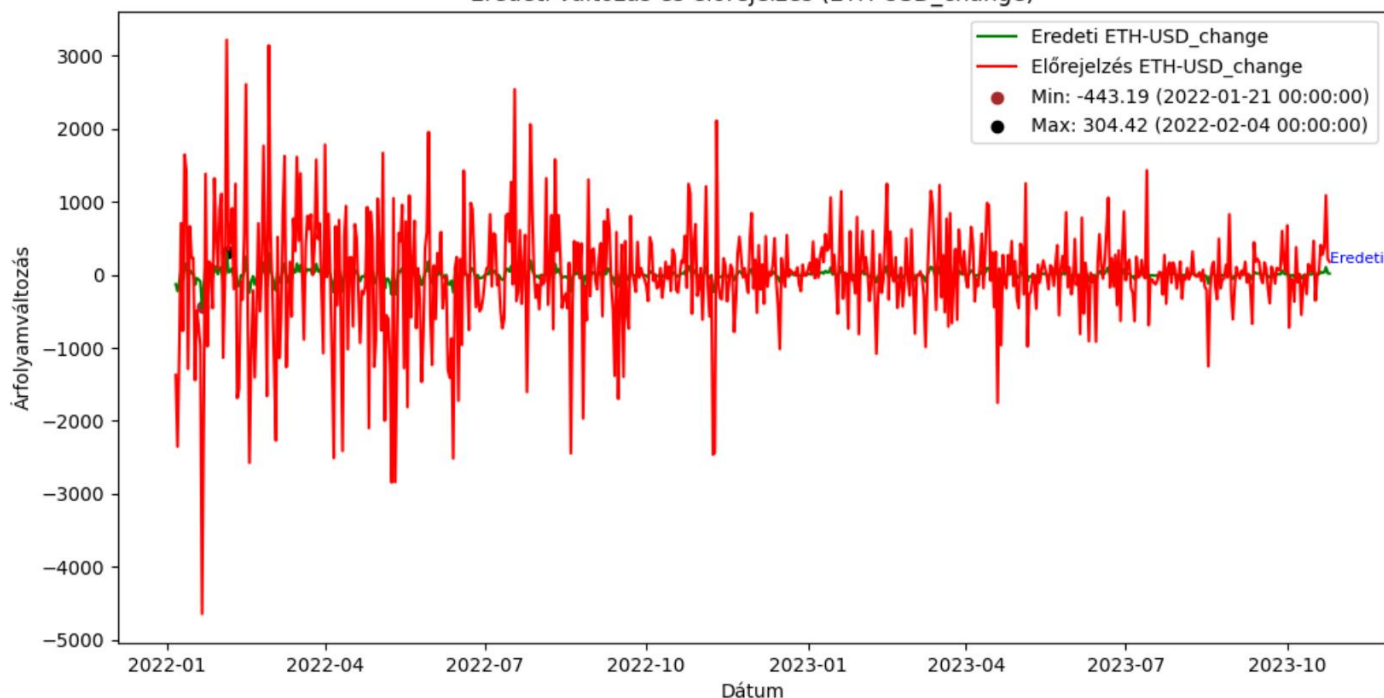
x_coord = df.index[-1]
y_coord = df['predict_LR'].tail(1).values[0]
predicted_today = df['predict_LR'].tail(1).values[0]

df_changed_for_label = df[ticker2+'_change'].tail(1).values[0]
label = f'Eredeti {ticker2}_change: {df_changed_for_label}, Predicted
Today: {predicted_today}'
plt.text(x_coord, y_coord, label, fontsize=8, color='blue')

plt.show()

```

Eredeti változás és előrejelzés (ETH-USD\_change)



Eredeti ETH-USD\_change: 15.50146484375, Predicted Today: 177.35837531417795

(napi valós változás a parancs futtatásáig, előre jelzett változás a napra)

A `model.predict()` szerint úgy kellett volna mozognia mint a piros vonalnak, pedig szinte alig történt lényeges árfolyammozgás. Az árfolyam előrejelzés sem helyes, ez lehet annak a hibája hogy a lineáris regresszió és a BTC-hez való hasonlítás nem egy jó stratégia, érdemesebb esetleg még pár fontos befolyásoló tényezőt belehelyezni a modellbe.

## Páros kereskedés (párhuzamos)

Mivel azt feltételezzük, hogy együtt mozognak legyen az a stratégia amikor a BTC nő akkor az ETH-ből veszünk, ha a BTC csökken akkor eladunk minden Ethereumot

```
ticker1_prices = df['Adj Close'][ticker1]
ticker2_prices = df['Adj Close'][ticker2]

# Stratégia létrehozása és backtesting
ticker2_position = 0
start_balance = 1000
initial_balance = start_balance # Kezdeti egyenleg

buy_pos_count = 0
sell_pos_count = 0

balances = [] # Itt tároljuk az egyenlegeket

for i in range(1, len(df)):
    ticker1_return = (ticker1_prices[i] - ticker1_prices[i - 1]) /
    ticker1_prices[i - 1] # Ticker1 napi hozam
    ticker2_return = (ticker2_prices[i] - ticker2_prices[i - 1]) /
    ticker2_prices[i - 1] # Ticker2 napi hozam

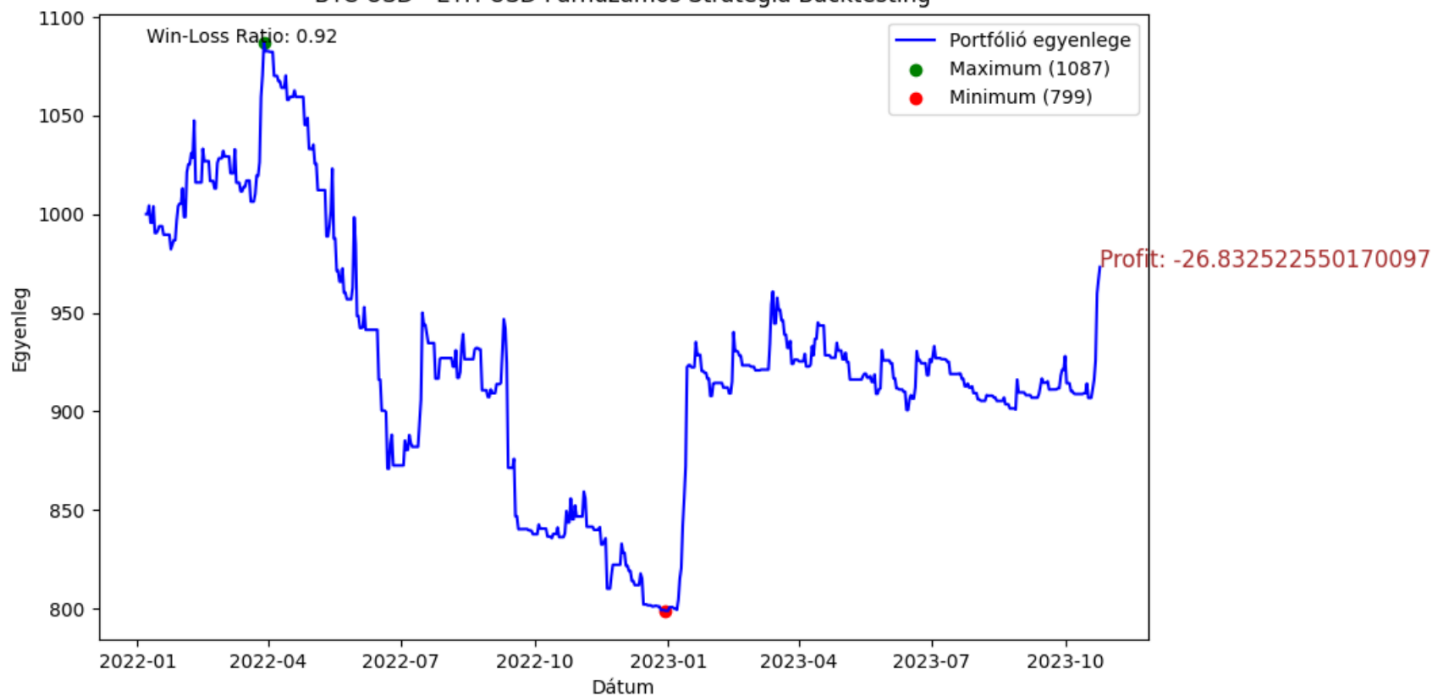
    if ticker1_return > 0:
        # Ticker1 növekedés esetén veszünk Ticker2-t
        ticker2_position += initial_balance * 0.2 /
        ticker2_prices[i] # 20% -ot veszünk Ticker2-ből
        initial_balance -= initial_balance * 0.2
        buy_pos_count = buy_pos_count + 1
    else:
        # Ticker1 csökkenés esetén eladunk Ticker2-t
        initial_balance += ticker2_position * ticker2_prices[i] # Az
        összes Ticker2 eladása
        ticker2_position = 0
        sell_pos_count = sell_pos_count + 1

    balance = initial_balance + ticker2_position * ticker2_prices[i]
    balances.append(balance)
```



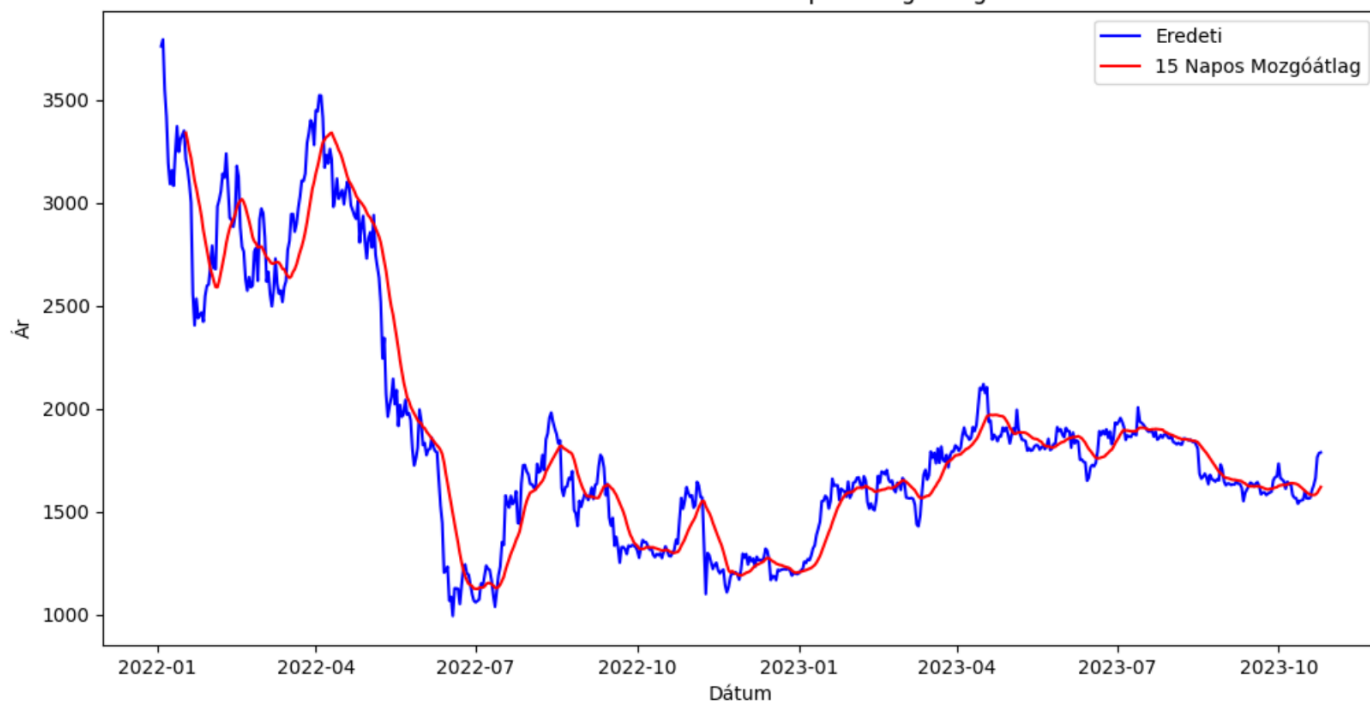
Vásárlás: 314 db  
Eladás: 343 db

### BTC-USD - ETH-USD Párhuzamos Stratégia Backtesting



BTC-ETH esetében -26\$ egyenlegben lennénk.

### ETH-USD - Eredeti és 15 Napos Mozgóátlag



2023.10.23

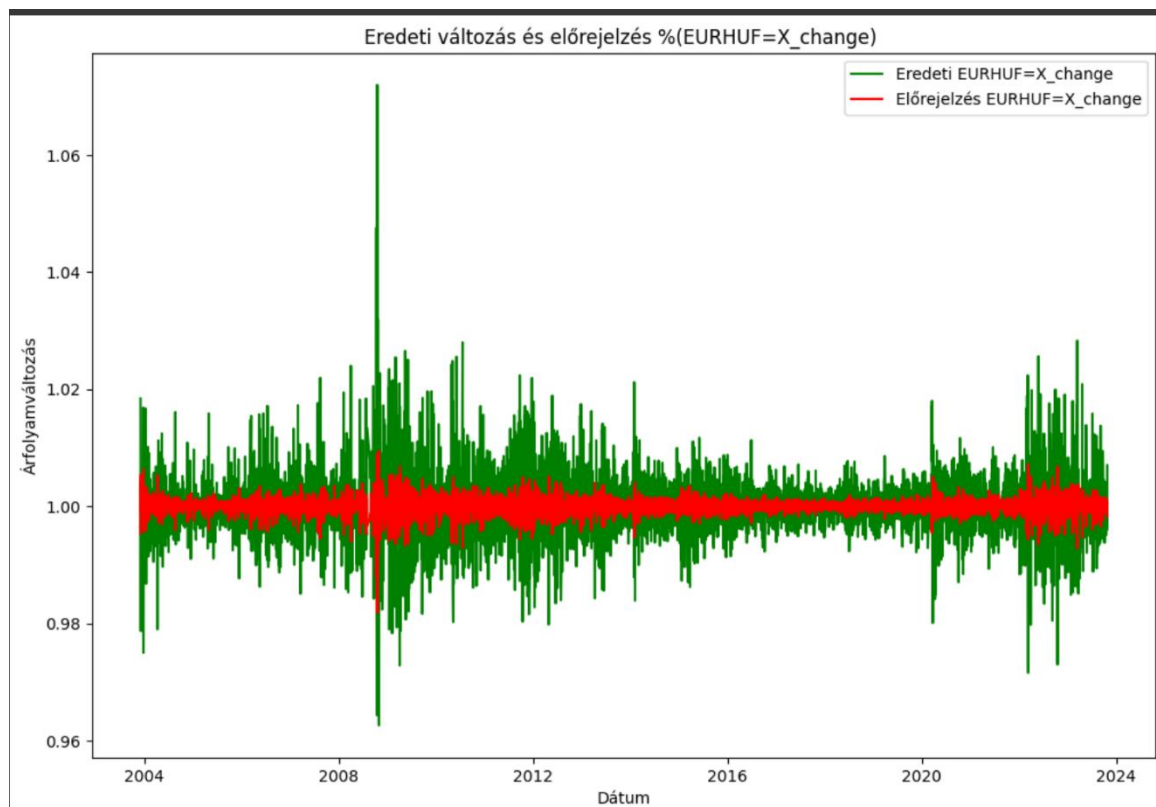
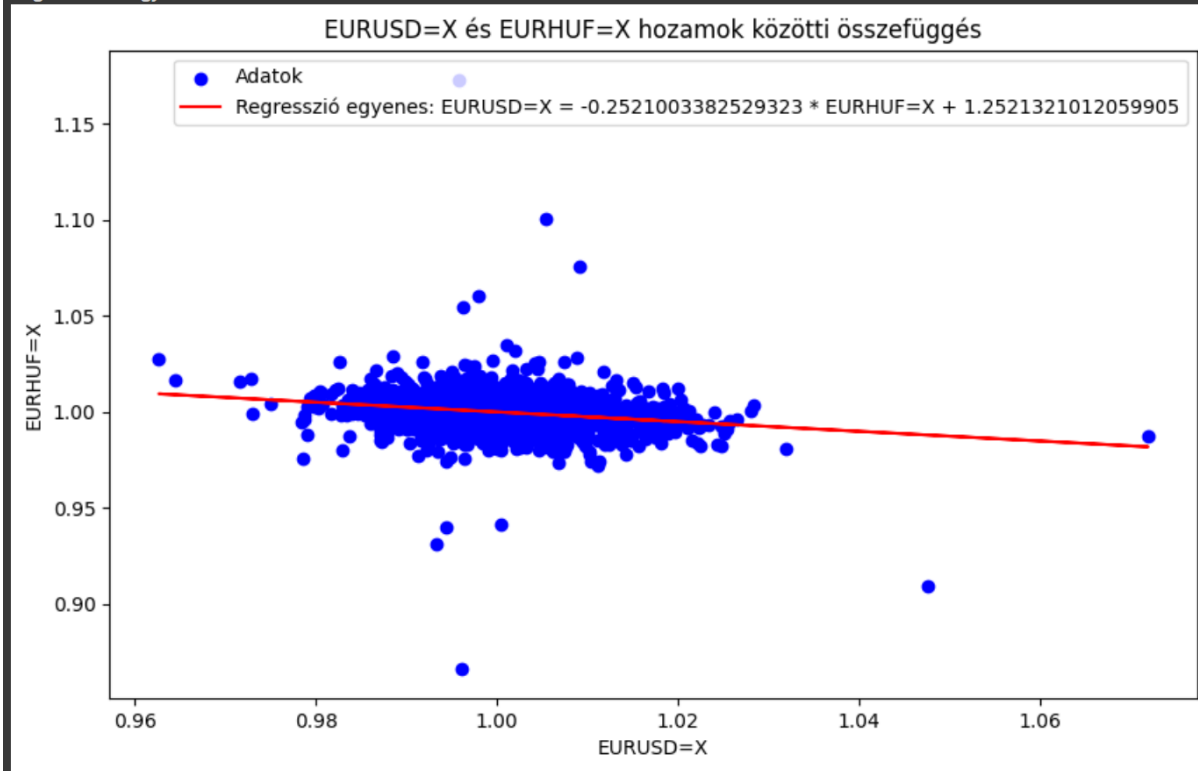
## Példák

### EURUSD=X / EURHUF=X

Korrelációs együttható: -0.1995

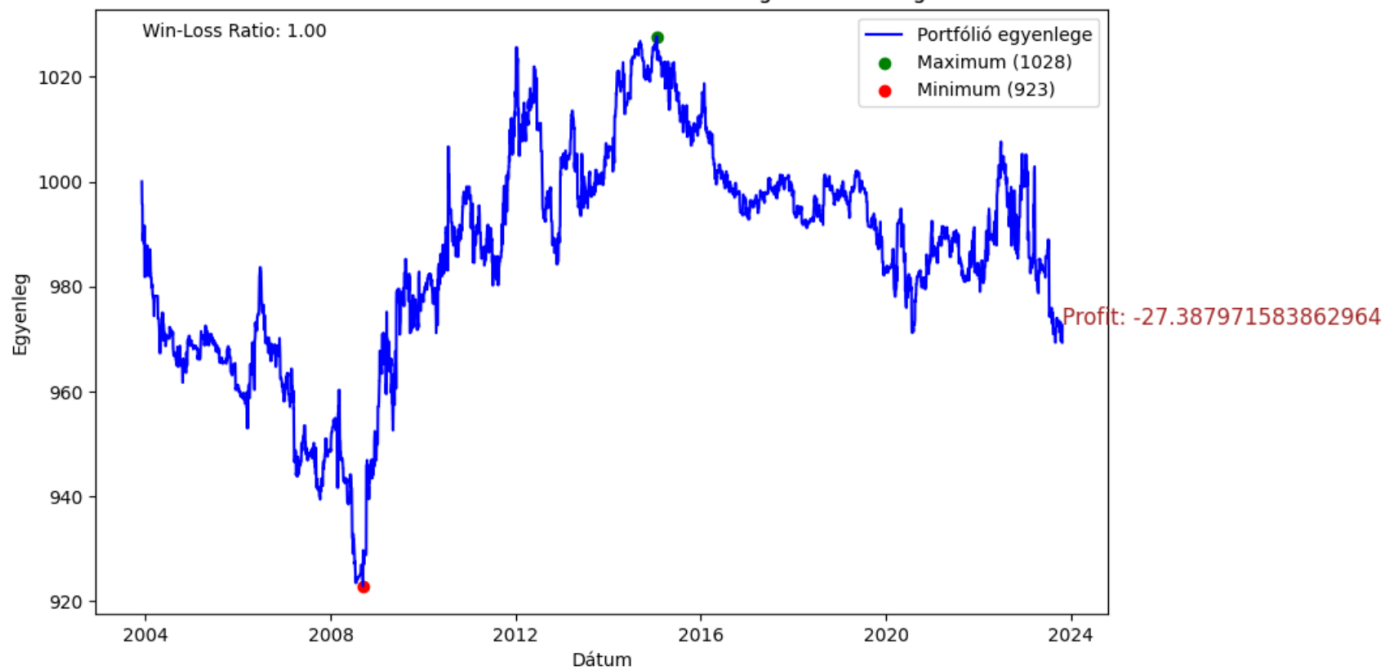
R-négyzet érték: 0.0398

Regresszió egyenes:  $\text{EURUSD}=\text{X} = -0.2521003382529323 * \text{EURHUF}=\text{X} + 1.2521321012059905$

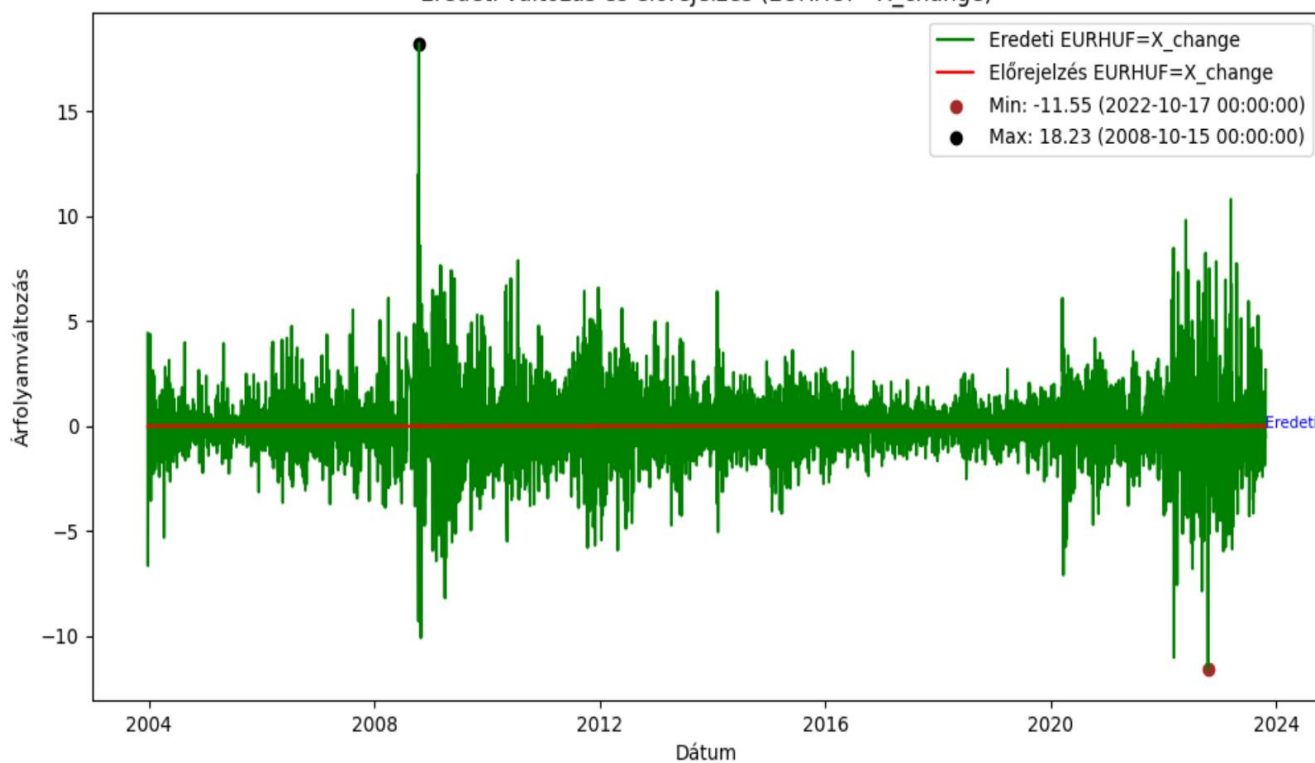


Vásárlás: 2580 db  
Eladás: 2578 db

### EURUSD=X - EURHUF=X Párhuzamos Stratégia Backtesting



### Eredeti változás és előrejelzés (EURHUF=X\_change)



## AAPL / MSFT

Korrelációs együttható: 0.4302

R-négyzet érték: 0.1851

Regresszió egyenes:  $AAPL = 0.5525402965616103 * MSFT + 0.44801527539089203$

