P04- Volatility-based statistical arbitrage on crypto

Team: Mykyta Luzan, Oleksandr Syzonov, Hermann Yavorskyi
Project owner: Federico Martinazzi (CHANGEHOLDING OÜ)
Cryptocurrencies are in demand as never before

- Mass adoption (El Salvador)
- USD inflation
- Still young and perspective market
- Trading benefits from bigger volatility in comparison with stocks
Key steps to build arbitrage automated bot

**Fundamental analysis**
Financial and crypto assets background research

**Statistical analysis**
Applying ML methods to predict price or public attitude

**Trading strategy**
Set of rules to extract profit from the previous analysis
Approach 1: Predict price

Steps:

1. Collect historical data
2. Split it to ranges
3. Train and evaluate on each range
4. Convert price prediction to binary value (is_growing)
Approach 1: Predict price

Steps:

1. Collect historical data
2. Split it to ranges
3. Train and evaluate on each range
4. Convert price prediction to binary value (is_growing)
Approach 1: Predict price

Steps:

1. Collect historical data
2. Split it to ranges
3. Train and evaluate on each range
4. Convert price prediction to binary value (is_growing)
Approach 1: Predict price

Steps:
1. Collect historical data
2. Split it to ranges
3. Train and evaluate on each range
4. Convert price prediction to binary value (is_growing)

Experiments:
1. Different prediction ranges (1, 5, 15, 30+ days)
2. Multidimensional features
3. Different scaling of the input
4. Applying logarithm to the values
Approach 1: Linear Regression

No log

RMSE: 648.069
Accuracy: 0.502

With log

RMSE: 706.722
Accuracy: 0.505
Approach 1: SVM

**No log**
- RMSE: 1052.345
- Accuracy: 0.471

**With log**
- RMSE: 536.756
- Accuracy: 0.476
Approach 1: Random Forest

No log

RMSE: 313.436
Accuracy: 0.497

With log

RMSE: 312.273
Accuracy: 0.513
Approach 2: Predict direction

Steps:
1. Collect historical data
2. Preprocess input to price differences
3. Split it to ranges
4. Train and evaluate classification on each range
Approach 2: Predict direction

Steps:

1. Collect historical data
2. Preprocess input to price differences
3. Split it to ranges
4. Train and evaluate classification on each range
Approach 2: Predict direction

Steps:
1. Collect historical data
2. Preprocess input to price differences
3. Split it to ranges
4. Train and evaluate classification on each range
Approach 2: Predict direction

Steps:
1. Collect historical data
2. Preprocess input to price differences
3. Split it to ranges
4. Train and evaluate classification on each range

Experiments:
1. Different prediction ranges (1, 5, 15, 30+ days)
2. Different scaling of the input
3. Applying logarithm to the values
## Approach 2: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Train Accuracy</th>
<th>Test Accuracy</th>
<th>Train F1-score</th>
<th>Test F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.566</td>
<td>0.516</td>
<td>0.705</td>
<td>0.666</td>
</tr>
<tr>
<td>SVM</td>
<td>0.763</td>
<td>0.545</td>
<td>0.820</td>
<td>0.670</td>
</tr>
<tr>
<td>Random Forest</td>
<td>1.0</td>
<td>0.540</td>
<td>1.0</td>
<td>0.617</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.847</td>
<td>0.554</td>
<td>0.873</td>
<td>0.647</td>
</tr>
<tr>
<td>XGBoost</td>
<td>1.0</td>
<td>0.530</td>
<td>1.0</td>
<td>0.587</td>
</tr>
</tbody>
</table>
Approach 3: SARIMA

Accuracy: 0.494  F1-score: 0.507
Trading strategies (low risk)
Trading strategies (mid risk)
Trading strategies (high risk)
Final results

**Return**: 751.08 %

**Strategy Equity**

**Capacity**

**Select Chart**
- Drawdown
- Capacity
- Strategy Equity
- Benchmark
- Assets Sales Volume
- Exposure

**Alpha Ranking**
- 37.8%
- Your algorithm is in the 42nd percentile.

**Research Guide**
- 95 Backtests Remaining
- 5 Parameters Detected
- 9 Minutes Research
# Project roadmap and progress

<table>
<thead>
<tr>
<th>Steps</th>
<th>Week</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Problem analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Data understanding</td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Data collection</td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Infrastructure setup</td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Initial solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Model training and experimentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Bot strategy tuning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>8. Comparing different assets for bigger profit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
</tr>
<tr>
<td>9. Final solution and presentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Done**
- **Partially done**
Lessons

1. Data collection is not that easy
2. Keep in mind your hardware limitations
3. Preprocessing matters more than models
4. Crypto trading is still need some research