Comparing MOEX and Yahoo Data for Trading Models

June 15, 2025

1 What We're Doing

I needed to check if Yahoo Finance data could work for building machine learning models for short-term trading on the Russian MOEX market. To do this, I compared the `close` prices from both datasets over three days (March 25–27, 2025) using a statistical test called Kolmogorov-Smirnov (KS). The idea is to see if the data behaves similarly enough to use Yahoo as a stand-in.

2 The Data

- **MOEX**: 36 candles total, 12 per day (9:00–23:00, with some gaps). It has columns like `id`, `open`, `close`, `high`, `low`, `volume`, and `begin`. The `volume` column has missing values for IDs 1434, 1472, 1475, 1476. `close` prices range from 3110.3 to 3210.0. - **Yahoo**: 40 candles (14 on March 25, 13 on March 26–27 due to missing 23:00). Columns are `datetime`, `close`, `high`, `low`, `open`, `volume`. `close` prices range from 3021.5 to 3069.3. - **Alignment**: I trimmed Yahoo to match MOEX's 36 timestamps (9:00–20:00, 22:00–23:00) so both datasets have the same number of candles.

3 How I Did It

Here's the plan: 1. **Dealing with Gaps**: For MOEX, I filled missing `volume` values with the average for that day (e.g., 3.5 for March 25). Since we're only looking at `close` prices, which are complete, this didn't affect the main analysis. For Yahoo, I dropped candles (like 10:00) that didn't have a matching MOEX timestamp. 2. **Normalizing**: The `close` prices are on different scales, so I standardized them to have a mean of 0 and standard deviation of 1:

$$z = \frac{x - \text{mean}}{\text{std}}$$

3. **KS Test**: This test checks if the distributions are similar by comparing their cumulative distribution functions:

$$D = \max |F_1(x) - F_2(x)|$$

If the p-value is above 0.05, the distributions aren't significantly different.

4 The Code

I used Python to crunch the numbers:

```
import pandas as pd
import numpy as np
from scipy.stats import ks_2samp

# Load data
moex_data = pd.DataFrame({
    'id': [1430, 1431, ..., 1476],
    'close': [3115.6, 3179.1, ..., 3155.0],
    'begin': ['2025-03-25 09:00:00', ..., '2025-03-27 22:00:00']
})
yahoo_data = pd.DataFrame({
    'close': [3027.3, 3028.5, ..., 3068.6],
    'datetime': ['25-03-25 09:00:00', ..., '25-03-27 22:00:00']
})

# Align datasets
moex_data['datetime'] = pd.to_datetime(moex_data['begin'])
yahoo_data['datetime'] = pd.to_datetime(yahoo_data['datetime'], format='%
yahoo_filtered = yahoo_data[yahoo_data['datetime'].isin(moex_data['dateti

# Standardize close prices
moex_close = (moex_data['close'] - moex_data['close'].mean()) / moex_data
yahoo_close = (yahoo_filtered['close'] - yahoo_filtered['close'].mean())

# Run KS test
ks_stat, p_value = ks_2samp(moex_close, yahoo_close)
results = []
for date in ['2025-03-25', '2025-03-26', '2025-03-27']:
    moex_day = moex_data[moex_data['datetime'].dt.date == pd.to_datetime(
    yahoo_day = yahoo_filtered[yahoo_filtered['datetime'].dt.date == pd.t
    moex_day_close = (moex_day['close'] - moex_day['close'].mean()) / moe
    yahoo_day_close = (yahoo_day['close'] - yahoo_day['close'].mean()) /
    ks_stat_day, p_value_day = ks_2samp(moex_day_close, yahoo_day_close)
    results.append({'Date': date, 'KS Statistic': ks_stat_day, 'P-Value':

results_df = pd.DataFrame(results)
```

5 What I Found

- **Overall (36 candles)**: KS Statistic = 0.1389, P-Value = 0.6723. Since p > 0.05, the distributions look similar. - **By Day**: All p-values are above 0.05, so no major

Date	KS Statistic	P-Value
2025-03-25	0.3333	0.3741
2025-03-26	0.2500	0.6827
2025-03-27	0.4167	0.1712

differences day by day either.

6 What It Means

The data suggests Yahoo's `close` prices follow a similar pattern to MOEX's after normalizing. This means Yahoo data could be used to train trading models for the Russian market. Just make sure to standardize the data and double-check the model on actual MOEX data to be safe. The small sample (36 candles) and short period (three days) are limitations, but the KS test handles this fine.

7 Wrapping Up

Yahoo's data looks close enough to MOEX's (p > 0.05) to be useful for short-term trading models. Standardize the prices, align timestamps, and validate on MOEX data for best results.