01 · Feature Engineering

Pull full-history SPX/USD 1-minute data, validate it, and persist engineered factors for downstream modeling.

Workflow

- Ensure `00_environment.ipynb` ran successfully
- Use HistData.com API to download SPX/USD M1 bars (cached under `data/prices.csv`)
- Compute momentum, volatility, and oscillator features
- Persist the final feature panel to `data/features.csv`

In [1]:
```
# Add parent directory to path for module imports
import sys
from pathlib import Path
sys.path.insert(0, str(Path.cwd().parent))

import pandas as pd

from momentum_lib import bootstrap_env, validate_prices, compute_features
from histdata_loader import ensure_histdata_prices

bootstrap_env(Path("../.env"))
print("Environment primed.")

data_dir = Path("../data")
data_dir.mkdir(exist_ok=True)
cache_file = data_dir / "prices.csv"
# Note: SPXUSD data on HistData.com is only available from 2019 onwards
# Earlier years will show warnings but won't affect the download
START_YEAR = 2019
# Set to False after first run to use cached data
REBUILD = True
```

```
Environment primed.
```

In [2]:
```
print("=" * 60)
print("LOADING PRICE DATA")
print("=" * 60)

# Note: First run downloads ~6 years of SPXUSD data (2019-2024) and caches zip file
# Subsequent runs reuse cached downloads and are much faster!
# To force re-download, delete data/histdata_cache/ directory

prices = ensure_histdata_prices(
    output_path=cache_file,
    start_year=START_YEAR,
    rebuild=REBUILD,
    verbose=True,  # Enable progress updates
)
prices = validate_prices(prices)
print(f"\n{'=' * 60}")
print(f"[OK] Data range: {prices.index.min()} to {prices.index.max()}")
print(f"[OK] Total rows: {len(prices):,}")
print(f"{'=' * 60}\n")
prices.head()
```

```
============================================================
LOADING PRICE DATA
============================================================
Rebuilding price data from 2019...
Using cache directory: ..\data\histdata_cache
[0001/0007]  14.3% Downloading SPXUSD 2019
[0002/0007]  28.6% Downloading SPXUSD 2020
[0003/0007]  42.9% Downloading SPXUSD 2021
[0004/0007]  57.1% Downloading SPXUSD 2022
[0005/0007]  71.4% Downloading SPXUSD 2023
[0006/0007]  85.7% Downloading SPXUSD 2024
[0007/0007] 100.0% Downloading SPXUSD 2025
Warning: Failed to download 2025: For the current year, please specify month=7 for e
xample.
For the past years, please query per year with month=None.
Building price data from 6 zip files...
Consolidating 6 data frames...
[OK] Price data saved to ..\data\prices.csv

============================================================
[OK] Data range: 2019-01-01 18:00:00-05:00 to 2024-12-31 16:12:00-05:00
[OK] Total rows: 1,943,357
============================================================
```

Out[2]:

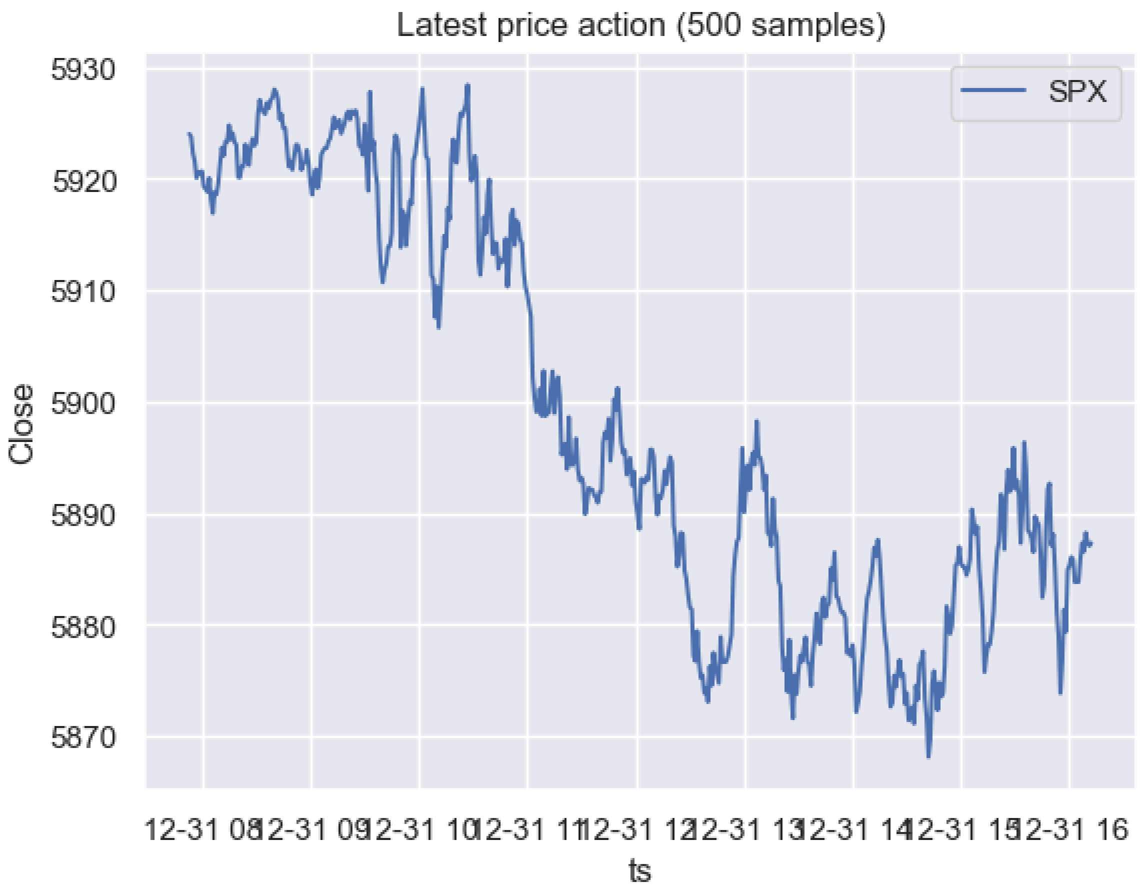
	SPX
ts	
2019-01-01 18:00:00-05:00	2508.070
2019-01-01 18:01:00-05:00	2511.269
2019-01-01 18:02:00-05:00	2515.270
2019-01-01 18:03:00-05:00	2514.270
2019-01-01 18:04:00-05:00	2514.270

In [3]:

```
# Import visualization libraries only when needed
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="darkgrid")

recent = prices.tail(500)
ax = sns.lineplot(data=recent)
ax.set_title("Latest price action (500 samples)")
ax.set_ylabel("Close")
```

```
plt.show()
recent.tail()
```



Out[3]:

	SPX
ts	
2024-12-31 16:08:00-05:00	5886.579
2024-12-31 16:09:00-05:00	5888.247
2024-12-31 16:10:00-05:00	5887.377
2024-12-31 16:11:00-05:00	5887.079
2024-12-31 16:12:00-05:00	5887.323

In [4]:

```
print("=" * 60)
print("COMPUTING FEATURES")
print("=" * 60)

features = compute_features(prices)

print(f"\n[OK] Features computed: {features.shape[1]} columns, {features.shape[0]:,

feature_file = data_dir / "features.csv"
print(f"Saving to {feature_file}...")
features.to_csv(feature_file, index=True)
```

```
print(f"[OK] Feature panel saved")
print(f"{'=' * 60}\n")

features.describe().T.head()
```

```
============================================================
COMPUTING FEATURES
============================================================
[OK] Features computed: 6 columns, 1,943,340 rows
Saving to ..\data\features.csv...
[OK] Feature panel saved
============================================================
```

Out[4]:

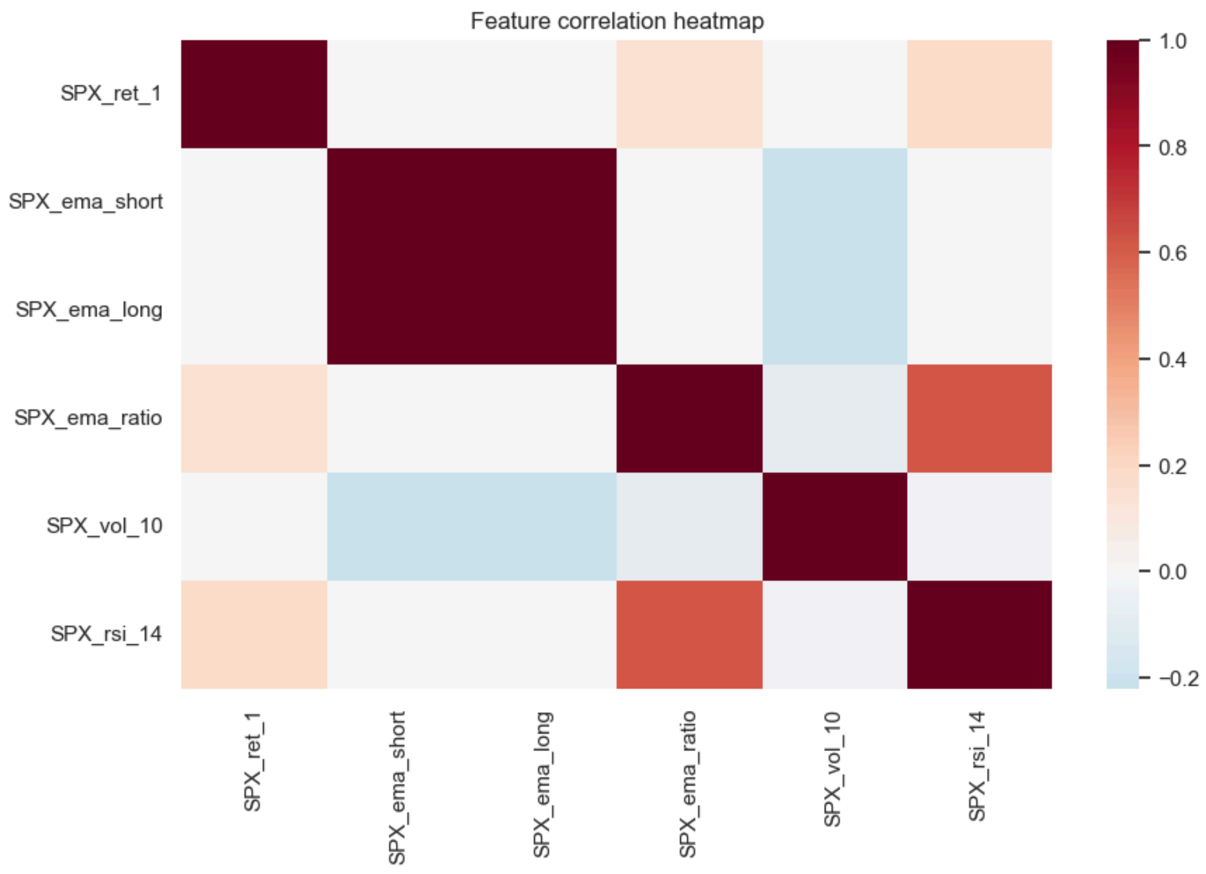
	count	mean	std	min	25%	50%
SPX_ret_1	1943340.0	4.995580e-07	0.000352	-0.073341	-0.000103	0.00000
SPX_ema_short	1943340.0	4.044457e+03	867.584138	2188.346939	3292.424784	4077.09204
SPX_ema_long	1943340.0	4.044444e+03	867.578505	2192.080701	3292.339003	4077.03297
SPX_ema_ratio	1943340.0	1.000003e+00	0.000521	0.963961	0.999856	1.00001
SPX_vol_10	1943340.0	2.211152e-04	0.000251	0.000000	0.000093	0.00015

In [5]:

```
# Ensure matplotlib is imported
if 'plt' not in dir():
    import matplotlib.pyplot as plt
    import seaborn as sns

print("Generating correlation heatmap...")
corr = features.corr(numeric_only=True)
plt.figure(figsize=(10, 6))
sns.heatmap(corr, cmap="RdBu_r", center=0)
plt.title("Feature correlation heatmap")
plt.show()
```

```
Generating correlation heatmap...
```


Feature correlation heatmap
SPX_ret_1
SPX_ema_short
SPX_ema_long
SPX_ema_ratio
SPX_vol_10
SPX_rsi_14
1.0
0.8
0.6
0.4
0.2
0.0
−0.2