02 · Modeling & Signal Discovery

Train a gradient boosted classifier that predicts whether SPX/USD will trade higher or lower over the next few minutes.

Steps

- Load features and aligned prices from `data/`
- Create future-return classification labels
- Train/validate an XGBoost model with scaling & imputation
- Surface performance metrics and feature importances

In [1]:

```python
from pathlib import Path
import sys

import pandas as pd

PROJECT_ROOT = Path("..").resolve()
if str(PROJECT_ROOT) not in sys.path:
    sys.path.append(str(PROJECT_ROOT))

from momentum_lib import (
    bootstrap_env,
    label_future_returns,
    train_model,
)

bootstrap_env(Path("../.env"))
data_dir = Path("../data")
features = pd.read_csv(data_dir / "features.csv", parse_dates=[0], index_col=0)
prices = pd.read_csv(data_dir / "prices.csv", parse_dates=[0], index_col=0)
aligned = features.index.intersection(prices.index)
X = features.loc[aligned]
y = label_future_returns(prices.loc[aligned], horizon=5)
X = X.loc[y.index]
print(X.shape, y.shape)
```

```
(1943340, 6) (1943340,)
```

In [2]:

```python
model, report, importances = train_model(X, y)
report
```

```
Out[2]: {'0': {'precision': 0.7299081206696217,
          'recall': 0.7206201006946658,
          'f1-score': 0.7252343741648583,
          'support': 188292.0},
         '1': {'precision': 0.7405706902333655,
          'recall': 0.7494260789715336,
          'f1-score': 0.7449720698105906,
          'support': 200376.0},
         'accuracy': 0.7354708903228462,
         'macro avg': {'precision': 0.7352394054514937,
          'recall': 0.7350230898330996,
          'f1-score': 0.7351032219877245,
          'support': 388668.0},
         'weighted avg': {'precision': 0.7354051593733605,
          'recall': 0.7354708903228462,
          'f1-score': 0.7354100523856258,
          'support': 388668.0}}
```

```
In [3]: importances.head(10) if importances is not None else "No feature importances availa
```

Out[3]:

	feature	importance
5	SPX_rsi_14	0.478099
3	SPX_ema_ratio	0.282669
0	SPX_ret_1	0.189252
4	SPX_vol_10	0.033196
1	SPX_ema_short	0.011269
2	SPX_ema_long	0.005515

```
In [4]: import joblib

        model_path = data_dir / "uprx_model.joblib"
        joblib.dump(model, model_path)
        print(f"Saved model to {model_path}")
```

```
Saved model to ..\data\uprx_model.joblib
```