APPLIED MACHINE LEARNING FINAL PROJECT

Predicting Kalshi Bitcoin Hourly Contract Price Moves

Guanran Tai, Zeyuan Peng, Jeppe Sebastian Pihl, Mads Ulrik Malmos

All group members contributed equally.

PRESENTATION OVERVIEW

From market question to stress-tested evidence

1. Background

What the Kalshi hourly BTC contracts mean.

2. Preprocessing

How raw trades become supervised contract-time rows.

3. Modeling

Leakage-safe features, baselines, and model selection.

4. Results

Main metrics, confusion matrix, and heterogeneity.

5. Stress tests

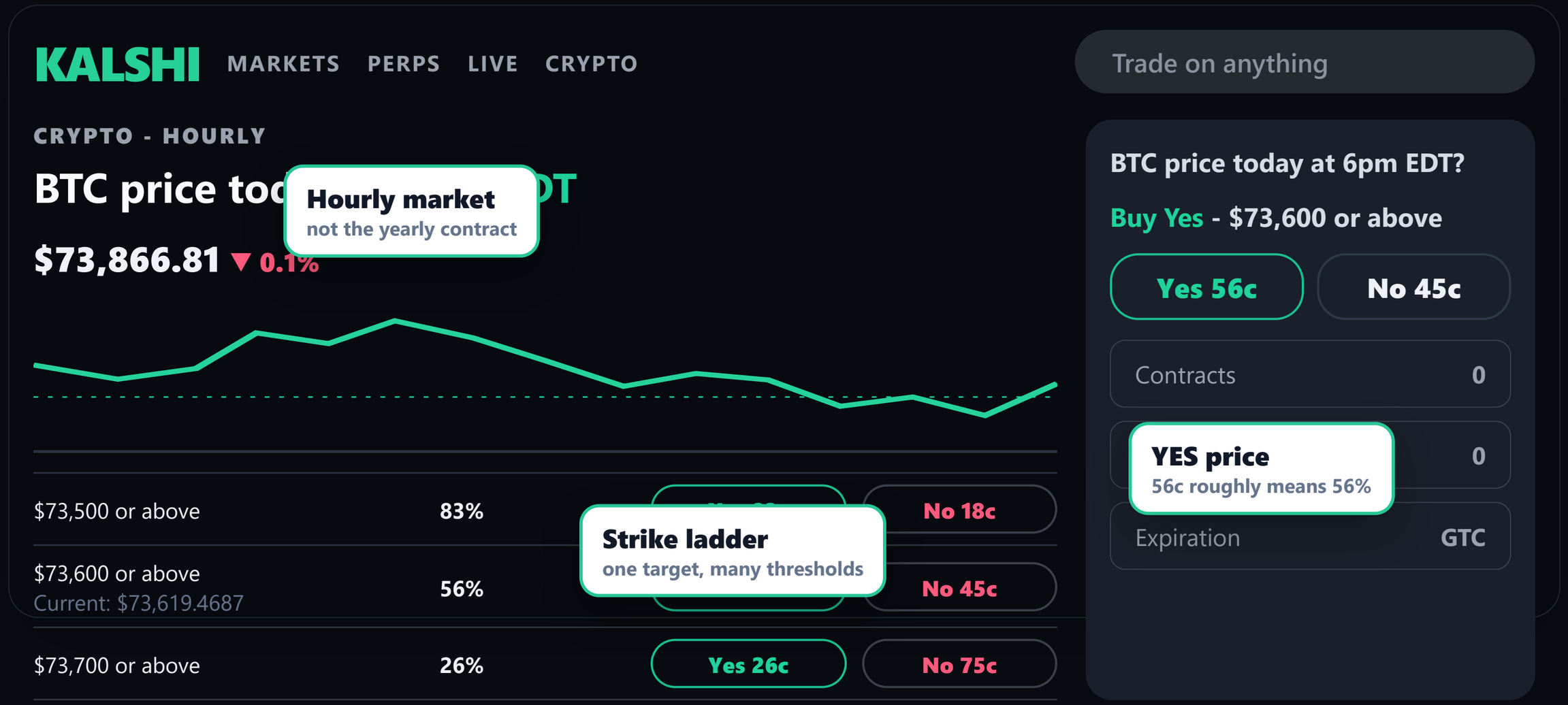
Class imbalance, ETH, NaNs, walk-forward, 10-minute data.

6. Conclusion

Predictive signal exists, but this is not yet a trading strategy.

BACKGROUND

A Kalshi hourly contract is a probability ladder



BACKGROUND

Our label is simply the next-minute movement of that YES price

At minute t

YES price is 56c.

->

One minute later

YES price is compared with the old price.

->

Target label

Down, unchanged, or up.

->

Features

Only information already known at t.

The ML problem is not "will BTC go up"; it is "will Kalshi's probability reprice in the next minute?"

BACKGROUND

Dataset: each row is one contract at one minute

Raw rows are trade events; supervised rows are contract-minute observations after resampling, target construction, and filtering.

219,966

train rows

82 contracts

62,849

validation rows

93 contracts

31,425

test rows

83 contracts

	Down	Unchanged	Up
Training	6.8%	86.2%	7.0%
Validation	10.0%	80.1%	9.9%
Test	16.3%	67.3%	16.4%

Imbalance problem: most YES prices are unchanged from one minute to the next.

Raw Kalshi trades: 949,316; raw markets: 10,000. These do not equal supervised rows by design.

Class distribution after the chronological 70/20/10 split. Class_0 = down, Class_1 = unchanged, Class_2 = up.

PREPROCESSING

Raw trades become fixed-frequency contract-minute rows

1-min resampling

For each contract-minute, keep the last observed Kalshi YES trade price.

Forward-fill

Inactive minutes inherit the previous YES trade price; engineered NaNs are filtered before main training.

Leakage removal

BTC and Kalshi activity features use information available at or before minute t.

Rolling features

Lagged returns/volume, volatility, SMA/EMA deviations, RSI, and Kalshi activity windows.

Derived features

Moneyness, distance to strike, time-to-expiry, and cyclical calendar features.

Encoding/scaling

Ticker is kept for grouping, not integer-encoded. Scaled models use train-set global scaling, not per-ticker scaling.

SECTION 2 - MODELING

We model one contract-minute at a time

The pipeline joins BTC spot prices, Kalshi trades, and contract metadata into a supervised time-series classification dataset.

MODELING

The pipeline has four moving parts

BTC prices

OHLCV, returns, rolling volatility, momentum.

Kalshi trades

Last YES trade price per contract-minute.

Contract metadata

Strike, expiry, time-to-expiry, moneyness.

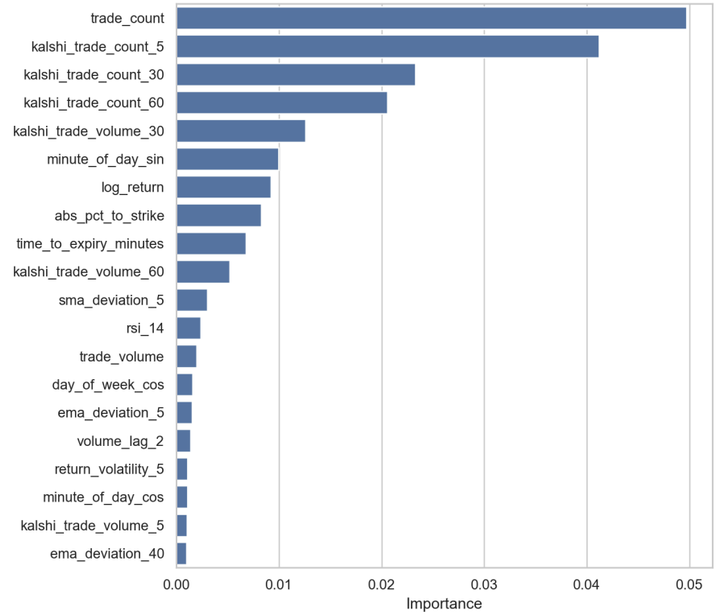
Label

YES price at t+1 vs YES price at t.

One row = one contract at one minute. One label = how its YES probability moves next minute.

MODELING

Features describe BTC movement, contract position, and market activity



Model tested: MLP. We use model's build-in feature importance measurement. Importance is sampled test-set permutation importance; [SHAP was considered but not added as a project dependency.]

60

numeric model features

68 total columns in model_data including ids, target, split, and timestamps.

BTC state

returns, lags, volume, volatility: what the underlying market is doing now.

Contract state

strike, expiry, moneyness: how close this contract is to becoming uncertain.

Kalshi activity

recent trade count, volume, time since last trade: whether this market is actively repricing.

MODELING

Hyperparameter optimization was limited and validation-only

Main model comparison

Each model family used a fixed, documented config chosen before test evaluation.

Manual + grid search

We did no random search and no Bayesian optimization.

XGBoost grid

12 configurations: 3 max_depth values x 2 learning rates x 2 tree counts.

Sequence models

Manual tuning: AdamW, hidden size, layers, dropout, learning rate, epochs, batch size.

Average scale

About 6 tunable hyperparameters per enabled model family in the config.

Selection rule

Best AUC wins; test metrics are reported once.

MODELING

Leakage control is the difference between a real result and a fake one

1

No future BTC

features use data at or before minute t

2

No YES price feature

target source is not handed to the model

3

70/20/10 split

actual supervised time range, not assumed calendar months

We choose the model using AUC, then report test results once.

MODELING

We compare simple baselines, classical ML, trees, and sequence models

Baselines
most frequent, stratified

Classical ML
logistic regression, MLP

Trees
LightGBM, XGBoost, Random Forest

Sequences
RNN, LSTM, GRU

Metrics
macro AUC, log loss, balanced accuracy, macro F1

Selection
AUC wins; tuning is validation-only

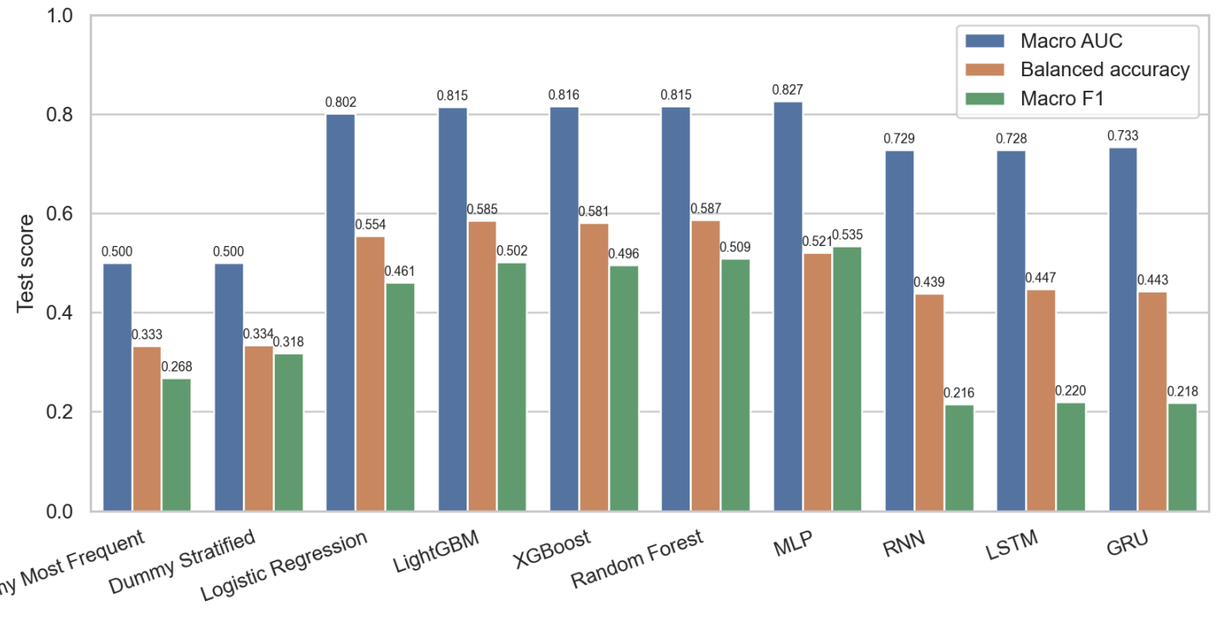
SECTION 3 - RESULTS

The corrected split still shows real predictive signal

Validation selects MLP, while Random Forest, XGBoost, and LightGBM remain close tree benchmarks.

RESULTS

Across models, the best results cluster among MLP and tree methods



MLP is selected by AUC, but the margin is small: 0.827 vs RandomForest 0.815. If optimizing balanced accuracy, RF is the stronger tree benchmark.

Figure shows test metrics for all trained models; model choice is based on AUC.

RESULTS

The final test result beats dummy baselines by a lot

0.827

Test AUC

macro one-vs-rest

0.521

Balanced accuracy

all classes count

0.535

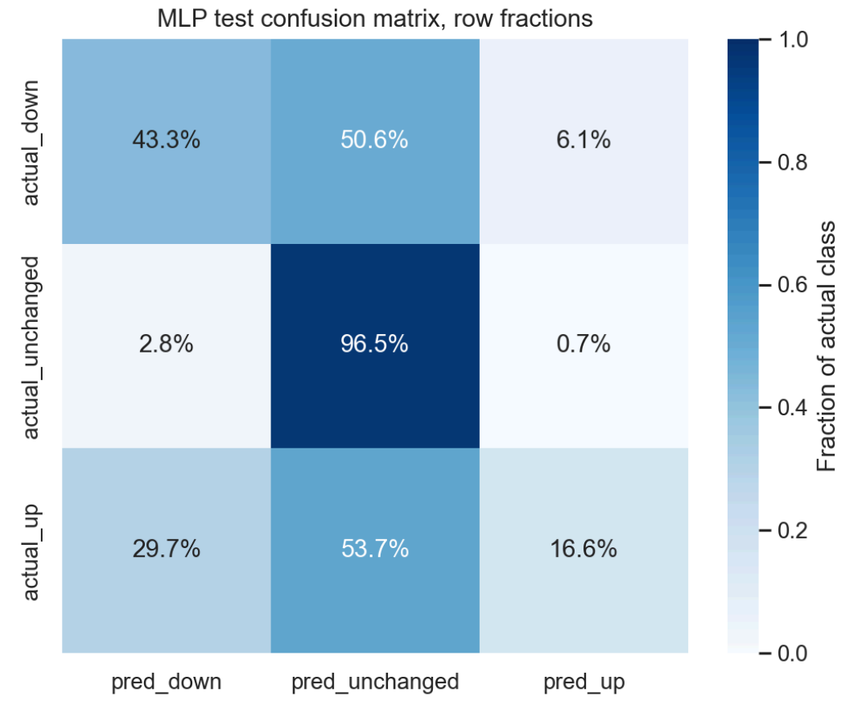
Macro F1

vs 0.268 dummy

Model tested: MLP. It ranks probability moves well, but class imbalance keeps macro F1 much lower than AUC.

RESULTS

The hard part is detecting moves without over-predicting them

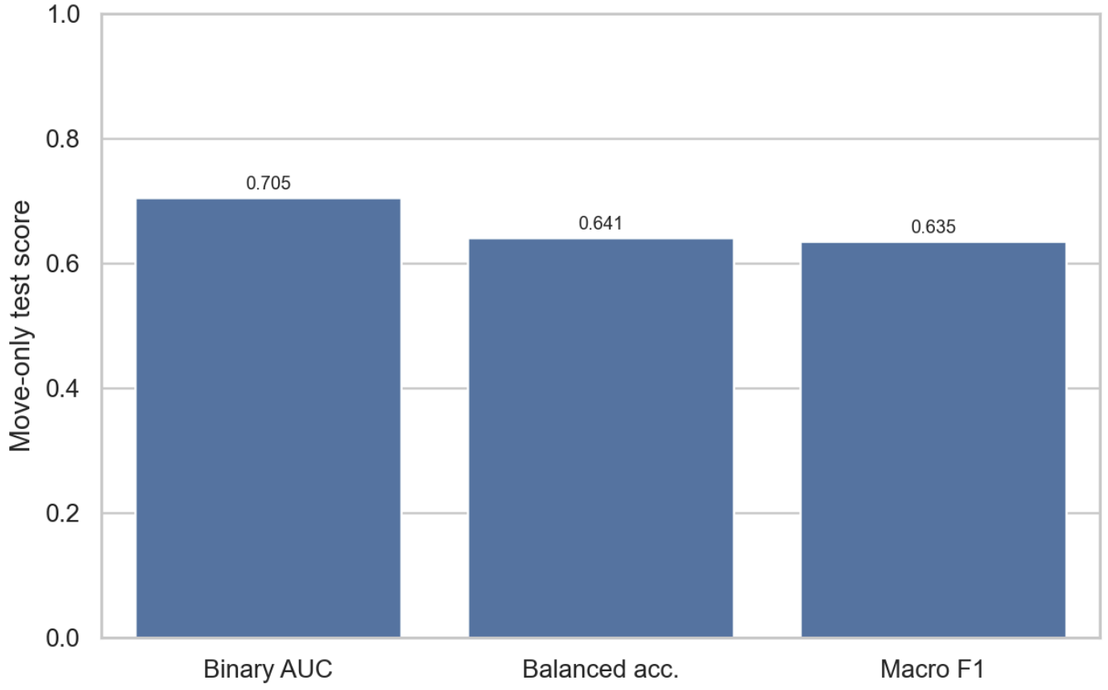


Unchanged is common, but the model still recovers some up/down moves. This is why macro metrics are the honest story.

Model tested: MLP. Cells are row fractions of each actual class.

RESULTS

When the market moves, we can also ask up vs down

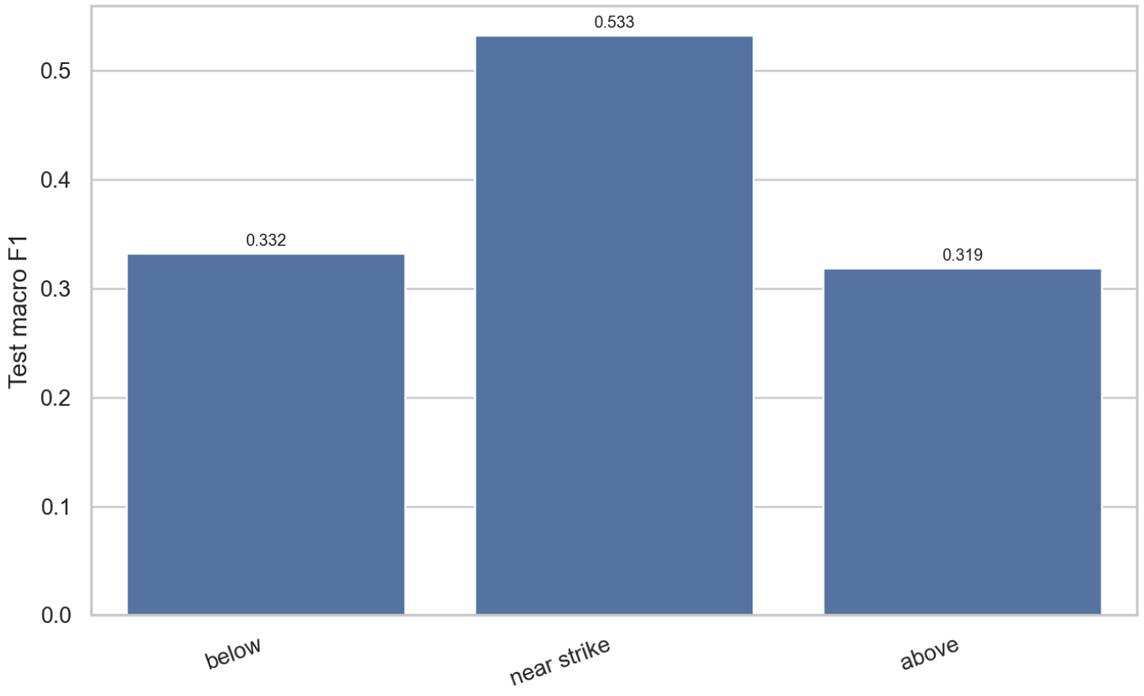


This diagnostic removes the unchanged class and asks a narrower question: conditional on a move happening, does the model rank up against down?

Model tested: MLP. Evaluation filters to rows where the true next-minute label is down or up.

RESULTS

Moneyness explains why some contracts are easier than others



Near-strike contracts are where probability can actually move. Far from strike, the market often stays unchanged.

Model tested: MLP.

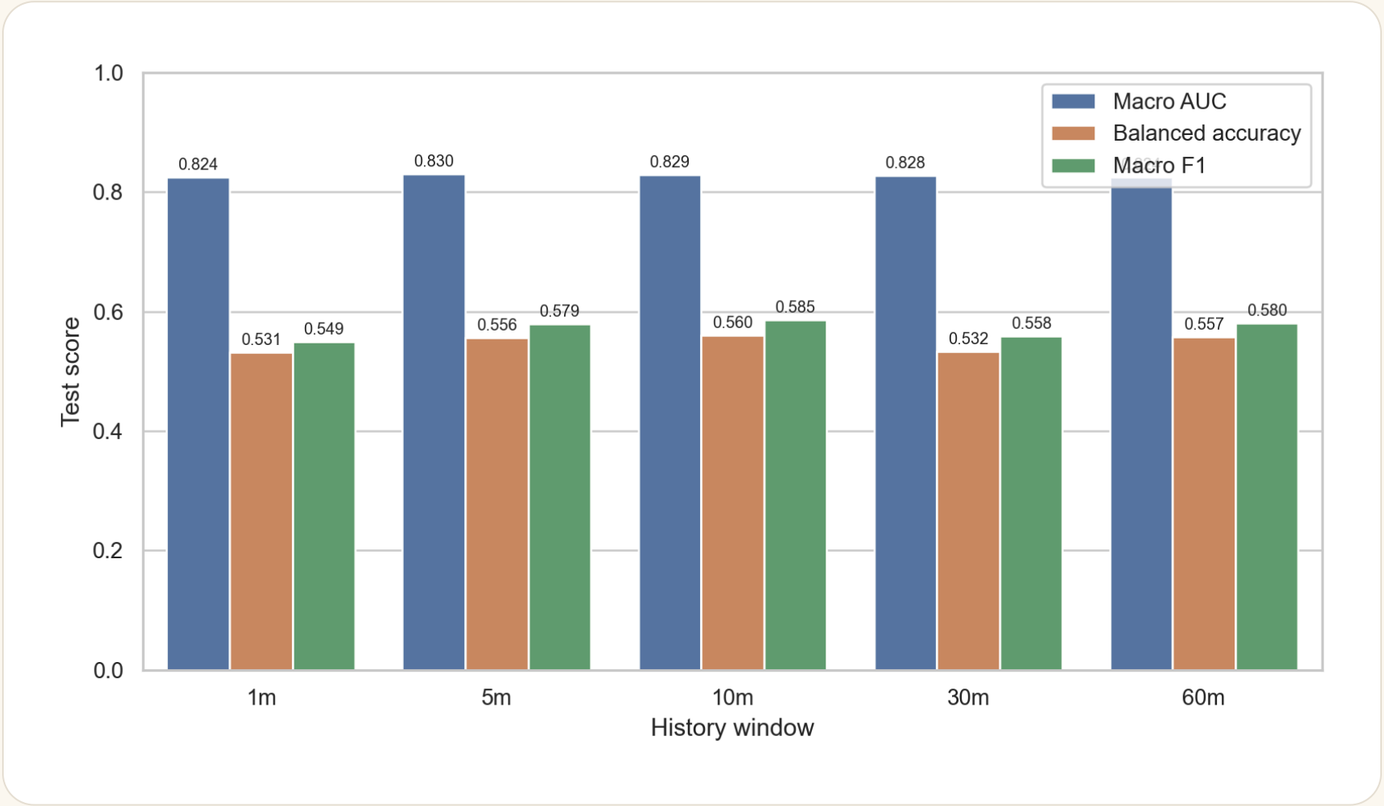
SECTION 4 - INTERESTING RESEARCH AREAS

The extra experiments test what the final model is actually using

For the main research section, each variant is trained separately with the AUC-selected model family: MLP.

RESEARCH AREA 1

One minute is already informative; longer context helps only slightly

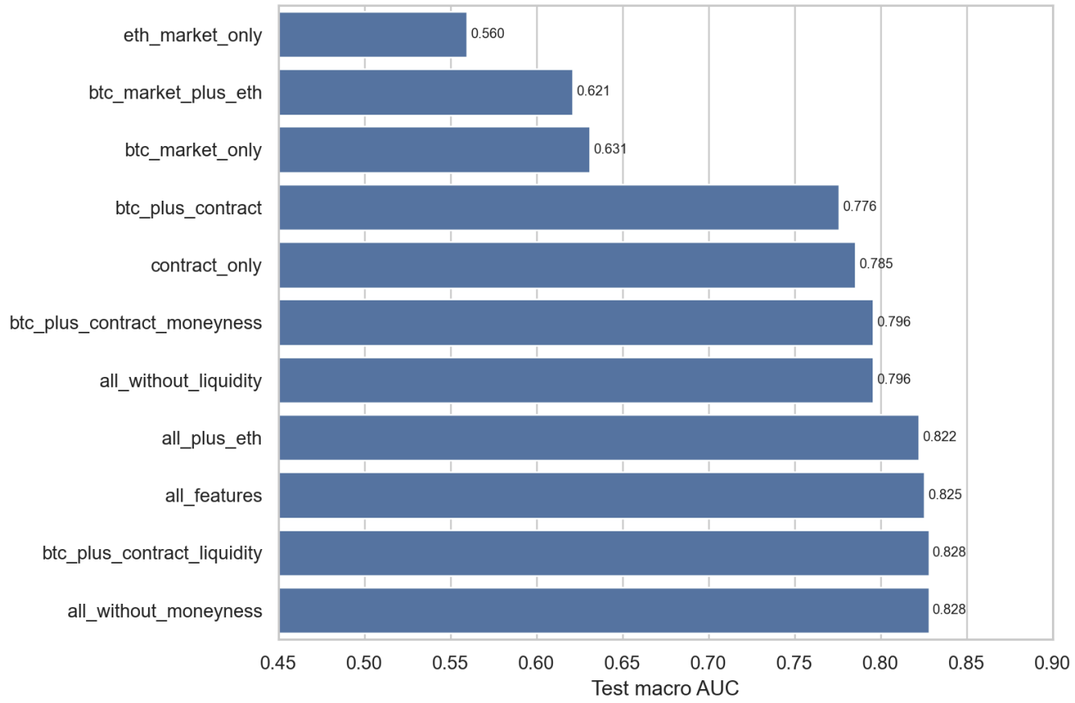


Our guess: next-minute repricing is mostly local. Current BTC level, moneyness, time-to-expiry, and Kalshi activity already summarize much of the short-term state, while longer lag windows are highly correlated with the current minute.

Model tested: MLP trained separately for each history window. The 1m variant has no lag/rolling/RSI lookback beyond current-minute state.

RESEARCH AREA 2

BTC-only features are not enough; contract activity carries signal

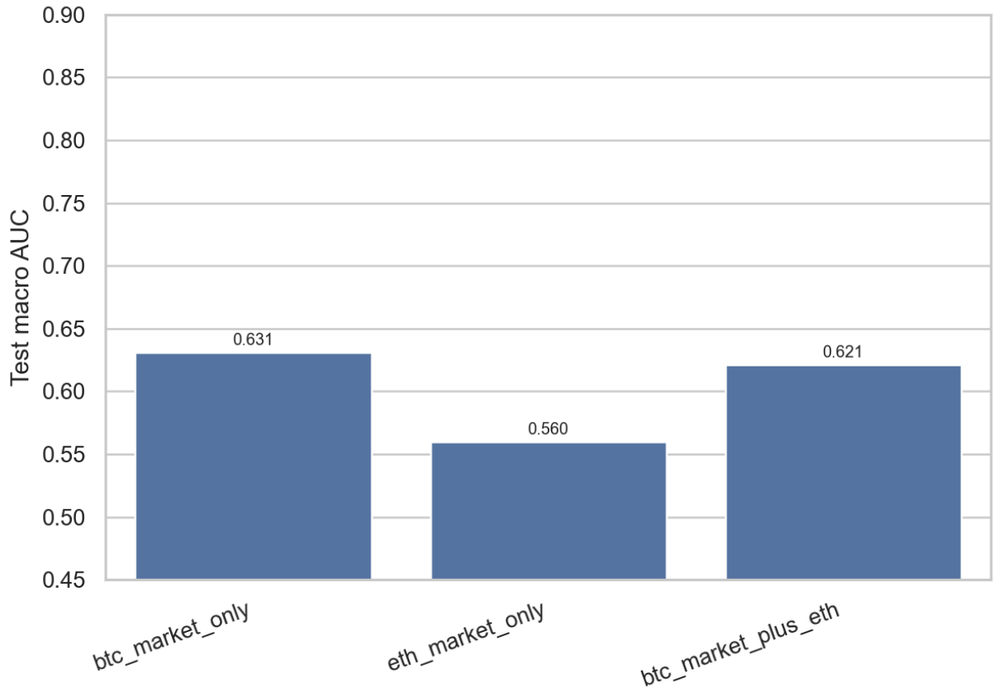


The useful signal is a mix of BTC movement, contract context, and Kalshi liquidity/activity.

Model tested: MLP trained separately for each feature-group variant.

RESEARCH AREA 3

ETH is highly correlated with BTC, but it adds little beyond BTC features



The ETH extension is a useful sanity check: cross-crypto movement is real, but BTC plus contract features already capture most of the predictive signal.

Model tested: MLP trained separately with BTC-only, ETH-only, and BTC+ETH feature groups.

CONCLUSION

Final answer: yes, but with boundaries

What worked

BTC + contract + activity features predict short-horizon probability moves.

What limits it

Most minutes are unchanged, and trade data miss quote-only updates.

Next step

Use order-book data, simulation, and transaction-cost-aware backtesting.

We can claim predictive signal. We should not claim a ready-to-trade strategy.

APPENDIX

Backup for technical questions

- Target construction and leakage controls
- Full metrics including log loss
- Sequence, volume, hyperparameter, expiry, and preprocessing checks
- Likely instructor questions and prepared answers
- All generated figures are catalogued at the end

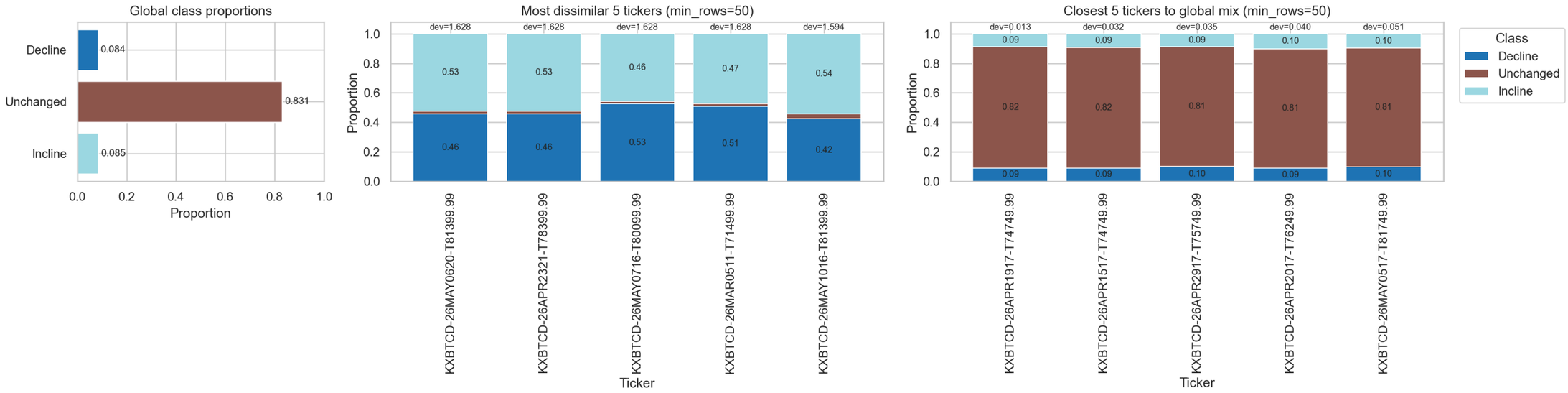
APPENDIX

Full test metrics

model	auc roc ovr macro	log loss	balanced accuracy	macro f1
dummy_most_frequent	0.500	11.775	0.333	0.268
dummy_stratified	0.500	14.303	0.334	0.318
logistic_regression	0.802	1.136	0.554	0.461
lightgbm	0.815	1.018	0.585	0.502
xgboost	0.816	1.026	0.581	0.496
random_forest	0.815	1.004	0.587	0.509
mlp	0.827	0.649	0.521	0.535
rnn	0.729	2.274	0.439	0.216
lstm	0.728	2.490	0.447	0.220
gru	0.733	2.223	0.443	0.218

APPENDIX

Ticker-level label mix can differ from the global imbalance



Descriptive diagnostic. Deviation is L1 distance between each ticker's class proportions and the global class proportions; tickers require at least 50 supervised rows.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Why is MLP selected instead of Random Forest?

Selection rule

We choose the final model using AUC, before reading test metrics.

Validation result

MLP has validation AUC 0.859; RandomForest has 0.855. The difference is real but small.

Test behavior

MLP has higher test AUC and lower log loss; RandomForest has higher balanced accuracy.

Interpretation

We report MLP as AUC-selected, but keep RandomForest as an interpretable tree benchmark and stress-test model.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Why do different sections use different models?

Final model

MLP is the headline model because it is selected by AUC.

Main research areas

History, feature-group, and ETH variants retrain MLP separately so they match the final model family.

Diagnostic models

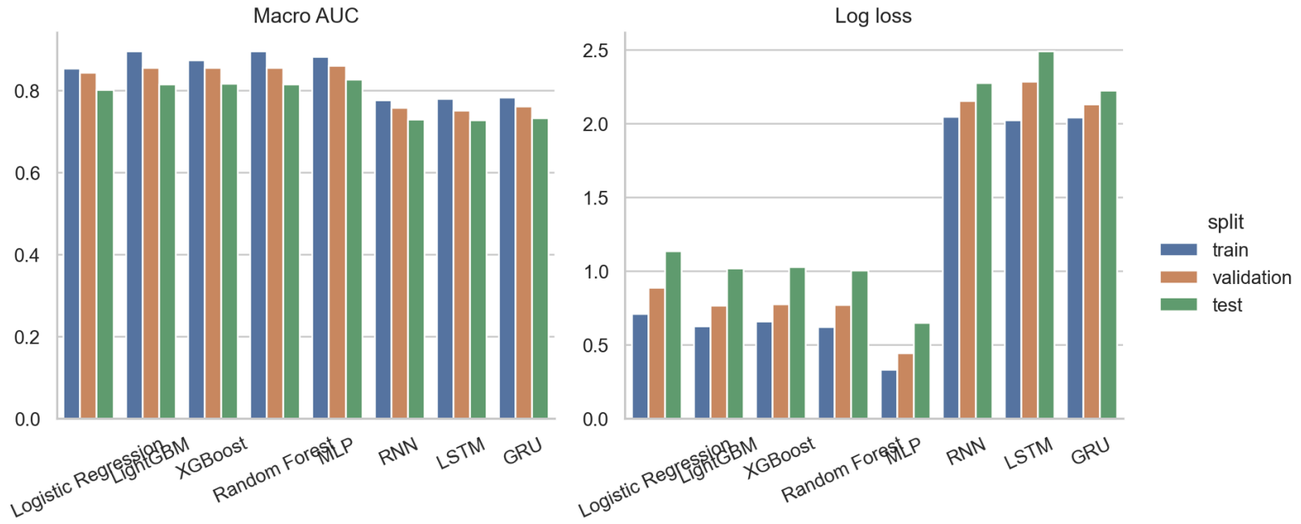
RandomForest and XGBoost are used for backup stress tests where fast repeated retraining is useful.

Logistic Regression

Used for preprocessing and NaN experiments because scaling and imputation choices are most visible in a linear pipeline.

APPENDIX

Sequence models became credible after tuning, but still did not win

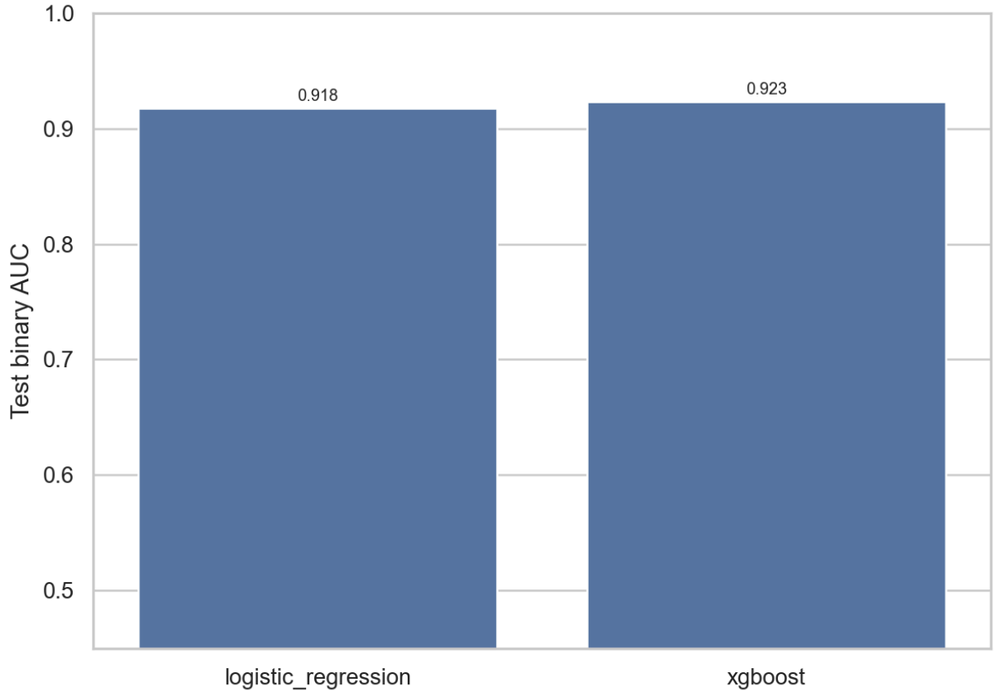


AdamW and stronger sequence settings improved RNN/LSTM/GRU baselines, but MLP remains the AUC-selected model.

Figure compares AUC and log loss across train, validation, and test for non-dummy models.

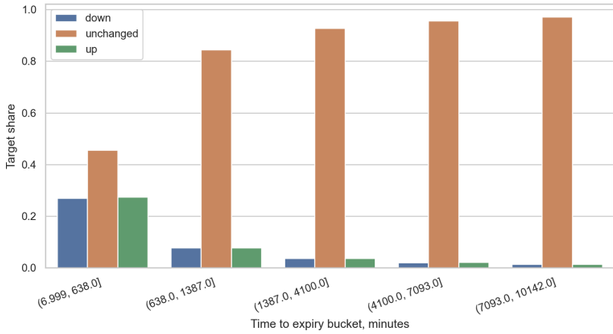
APPENDIX

Volume and expiry checks add useful context



Auxiliary task: predict high next-minute Kalshi volume, not YES-price direction.

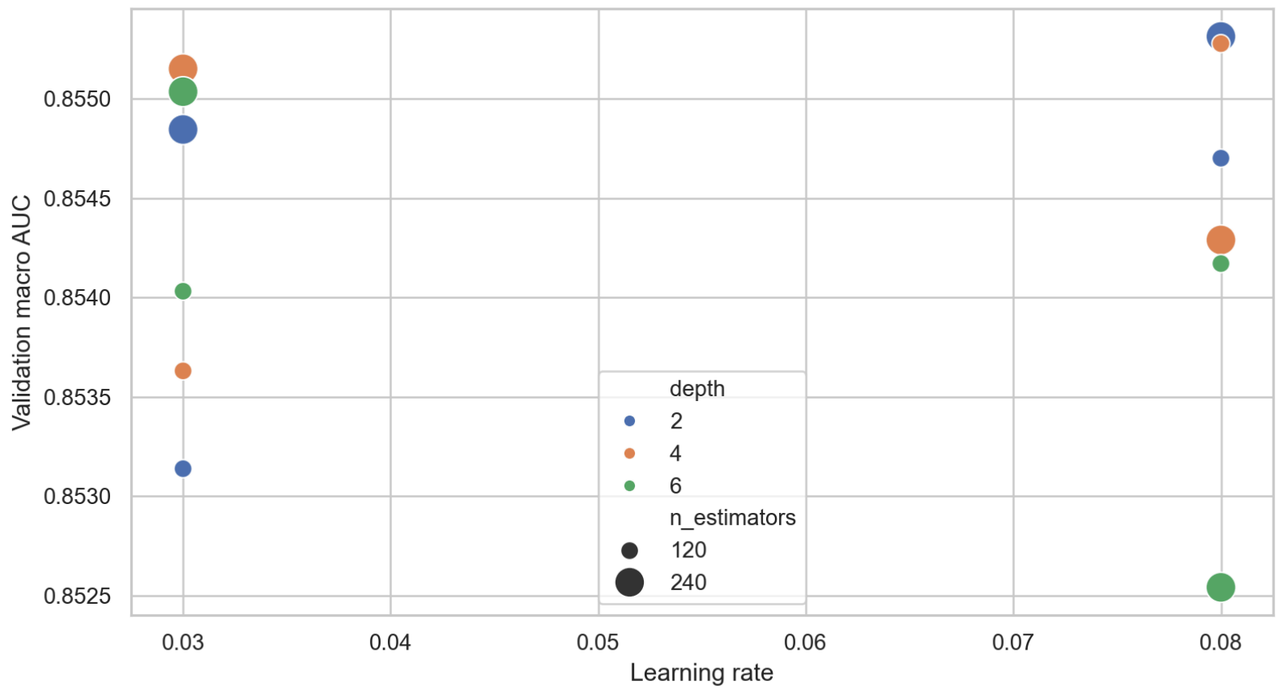
Volume is easier to predict than direction, which suggests liquidity/activity is a real signal channel.



Lower figure is descriptive target mix by time-to-expiry bucket.

APPENDIX

Hyperparameter search does not overturn the model story

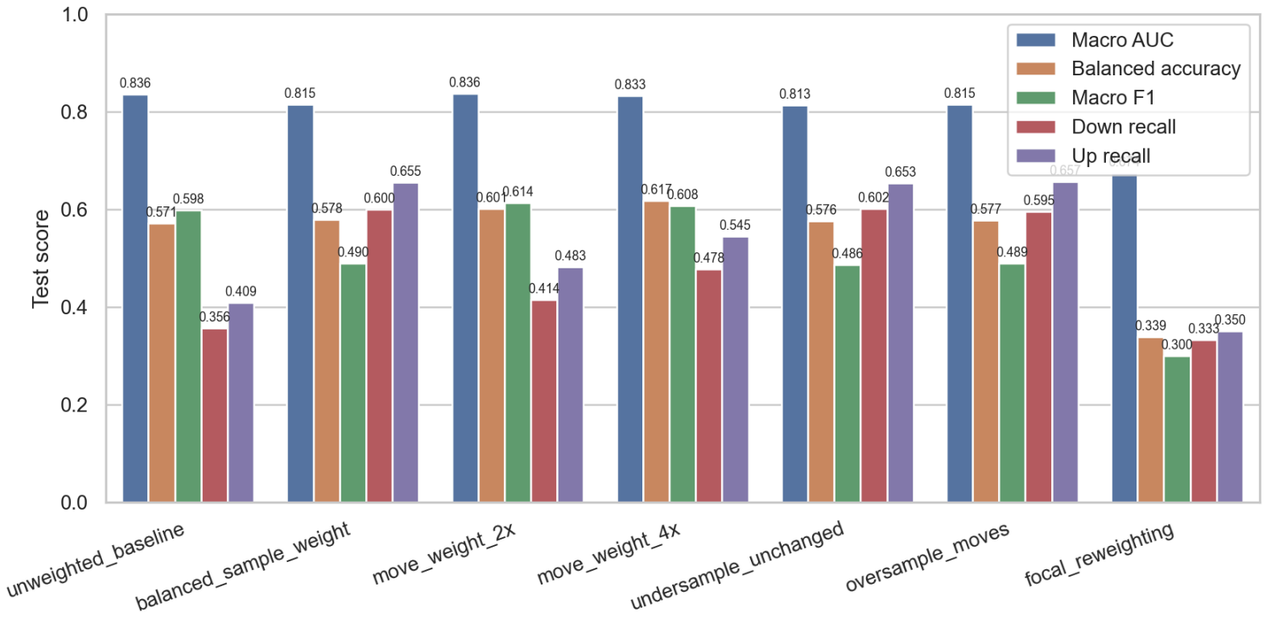


Small XGBoost search variants cluster near the same AUC range, so the conclusion is not driven by one arbitrary setting.

Model tested: XGBoost grid variants; selection is validation-only, test shown for comparison.

APPENDIX

Class weighting changes the move-class trade-off

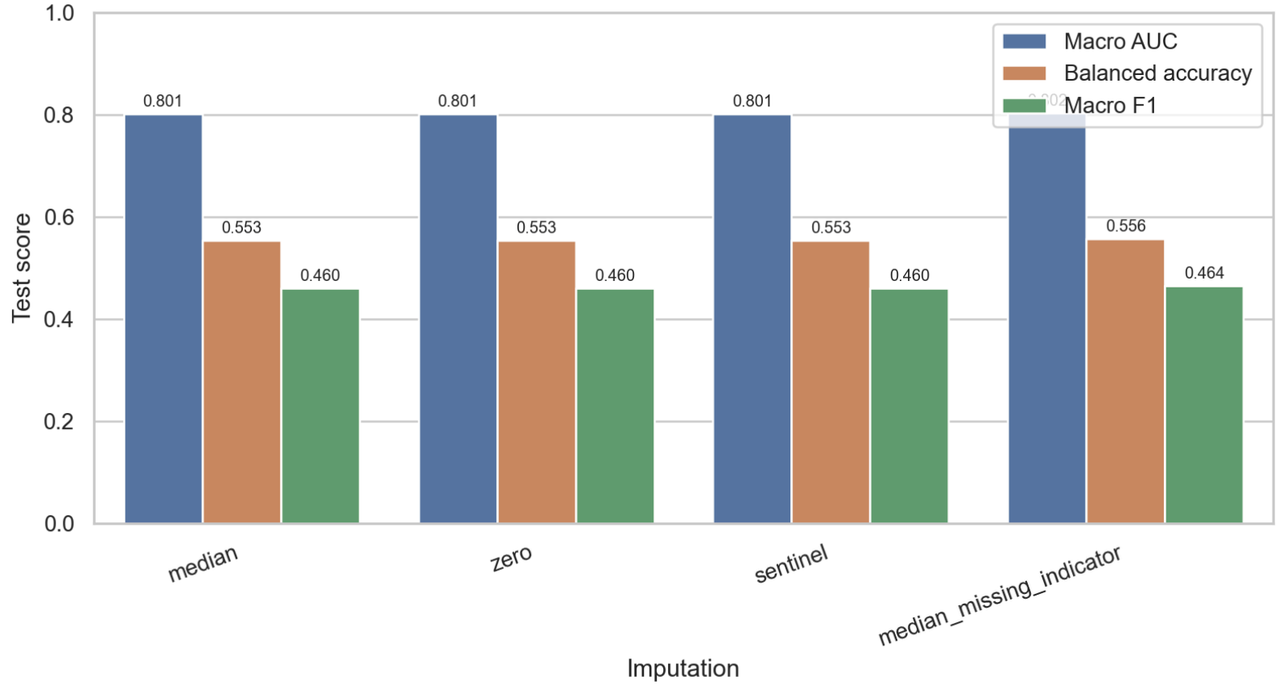


Move-class weighting is useful if we care more about up/down recall. It is not a free improvement for every metric.

Model tested: XGBoost variants. Diagnostic stress test, not the final model-selection rule.

APPENDIX

Sentinel NaN handling does not change the final conclusion

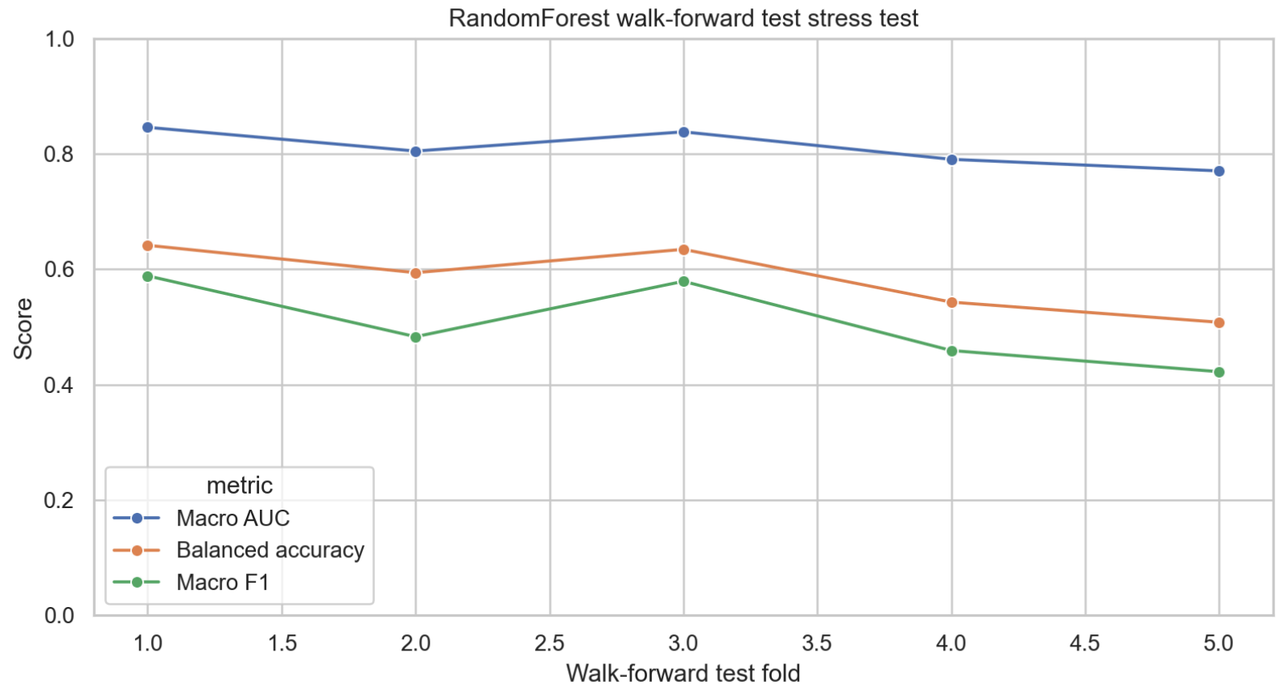


After final filtering, median, zero, sentinel, and missing indicators perform almost identically.

Model tested: Logistic Regression. Diagnostic preprocessing sensitivity check.

APPENDIX

Walk-forward testing shows performance changes across the test period

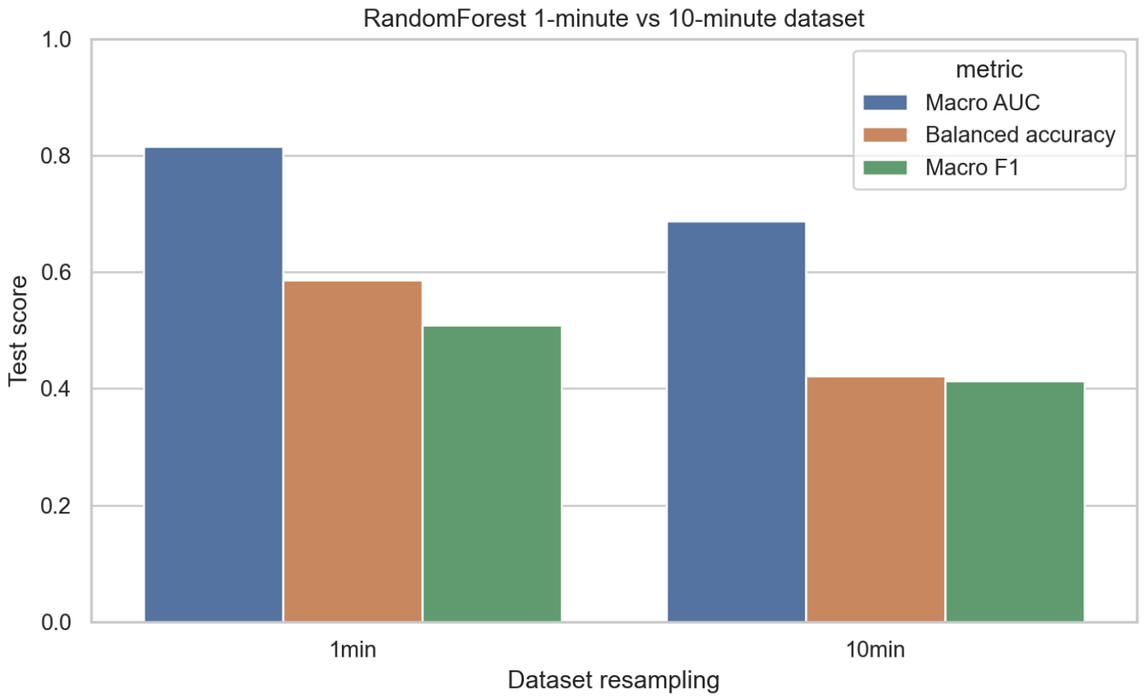


The folds remain above dummy performance, but the final folds are harder. That is a warning about time variation.

Model tested: RandomForest, five chronological test folds. Appendix robustness check.

APPENDIX

Resampling to 10 minutes loses short-horizon signal



The target is a rapid repricing problem. Coarser 10-minute bars smooth away information that matters at the next-move horizon.

Model tested: RandomForest. Main 1-minute RF test AUC 0.815; 10-minute RF test AUC 0.687.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

How do tree models see the past?

Lag columns

Past BTC values are explicit columns such as close_lag_1, close_lag_10, and log_return_lag_5.

Rolling summaries

Volatility, SMA/EMA deviations, RSI, and Kalshi activity windows summarize recent history.

Tree split logic

Trees do not remember sequences; they split on these static lag/rolling columns.

Sequence models

RNN/LSTM/GRU reconstruct short sequences from lagged tabular features, but they did not win validation AUC.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Is the target just measuring no-trade forward fill?

Concern

Kalshi trades are irregular. If we forward-fill inactive minutes, unchanged labels can partly mean "nothing traded."

What we did

We explicitly model this market state and report balanced/macro metrics so the unchanged majority does not hide move-class performance.

Evidence

Kalshi activity features are important, and the separate volume-prediction task reaches high AUC.

Limitation

A stronger next version should use order-book mid prices, not only trade prices.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Why exclude the current YES price from features?

Reason

The target is constructed from YES price movement. Feeding current YES price risks learning contract price levels instead of BTC-to-market transmission.

Design choice

We use BTC state, contract metadata, moneyness, and Kalshi activity, but not the current YES price itself.

Trade-off

This is conservative: it may remove useful information, but it makes the prediction claim cleaner.

Future work

A separate calibrated market-making version could include YES price and evaluate trading simulation.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Could train/test leakage come from overlapping contracts or time?

1

Time order

Train, validation, and test are chronological.

2

Row-time split

Each supervised row is assigned by timestamp over the actual available range.

3

Feature timing

BTC features use data available at or before minute t.

This makes the test period closer to a realistic future-period evaluation than a random row split.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Why do we emphasize AUC, balanced accuracy, and macro F1?

AUC

Measures whether predicted probabilities rank true classes well.

Balanced accuracy

Gives down, unchanged, and up equal influence.

Macro F1

Penalizes ignoring rare move classes.

Raw accuracy is dangerous here because the unchanged class is dominant.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Why did ETH not add much despite high BTC-ETH correlation?

Observation

BTC and ETH minute returns are highly correlated in this period.

Interpretation

Because the target is a BTC contract, BTC features and contract context already absorb most of the shared crypto-market movement.

Result

The all-plus-ETH variant is very close to the all-feature variant, not a step-change improvement.

Takeaway

ETH is a sanity check, not a new main signal channel.

APPENDIX: LIKELY INSTRUCTOR QUESTIONS

Can this be used as a trading strategy?

Short answer

No claim yet. The project shows predictive signal, not profitability.

Missing pieces

Order book, spread, fees, liquidity, fill probability, and transaction costs.

What the model gives

Probability ranking over next-minute direction.

Next experiment

Calibration plus a transaction-cost-aware backtest.

APPENDIX

Main limitations

Trade data, not full order book

Quote updates without trades are not fully captured.

No trading simulation

Transaction costs and fill risk are outside this project.

Limited hyperparameter search

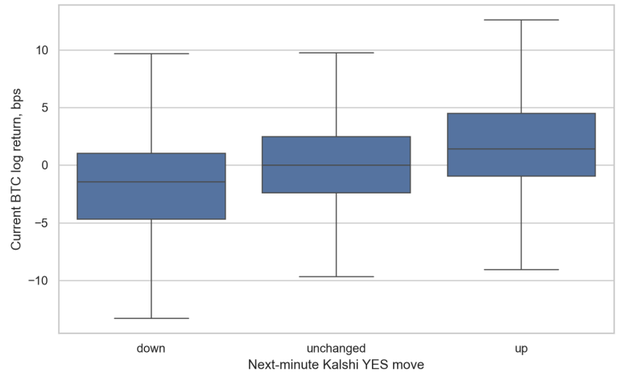
The focus is defensible comparison, not exhaustive tuning.

Time period risk

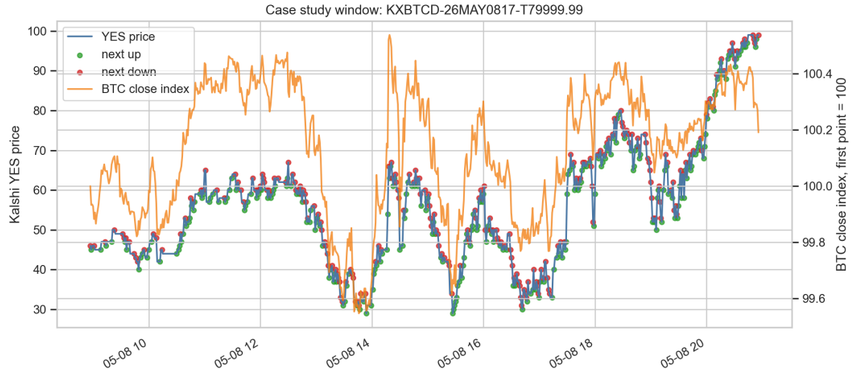
Walk-forward stress testing shows the final test folds are harder.

APPENDIX FIGURE CATALOG

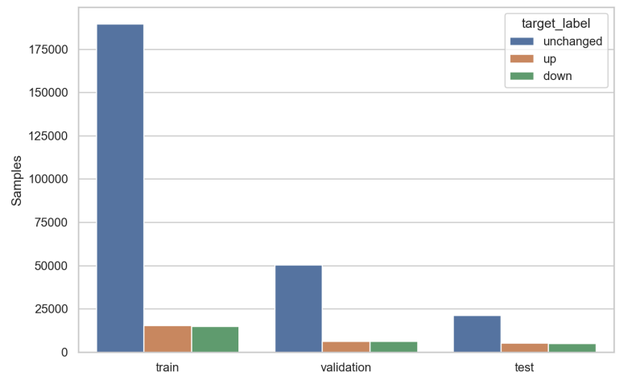
Generated figure backup



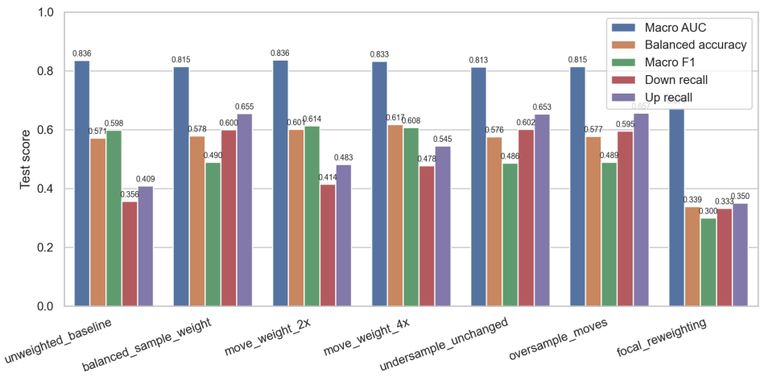
Btc Return By Target



Case Study Time Series



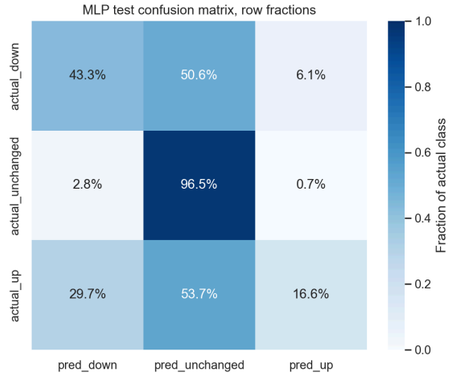
Class Distribution



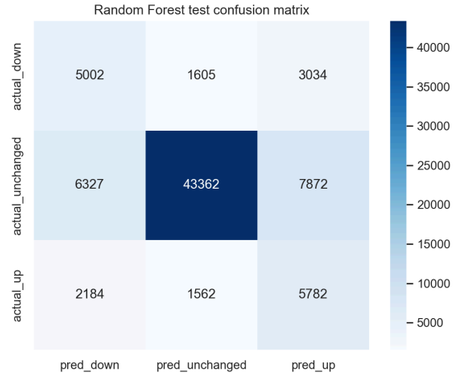
Class Imbalance Experiment

APPENDIX FIGURE CATALOG

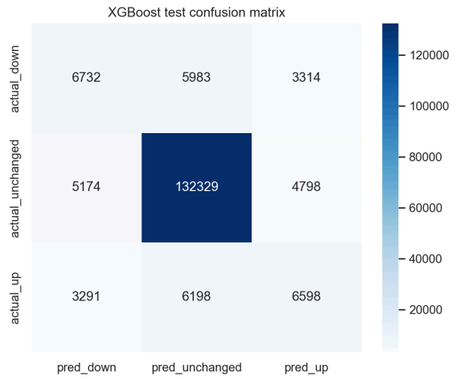
Generated figure backup



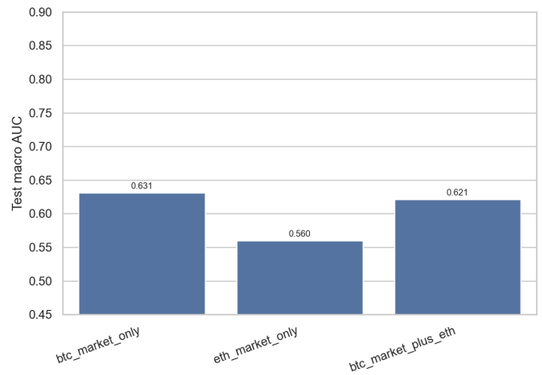
Confusion Matrix Mlp Test



Confusion Matrix Random Forest Test



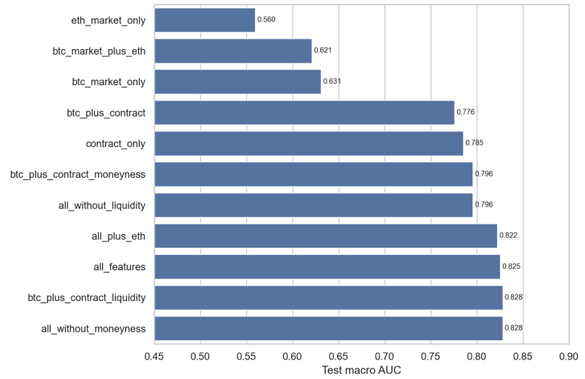
Confusion Matrix Xgboost Test



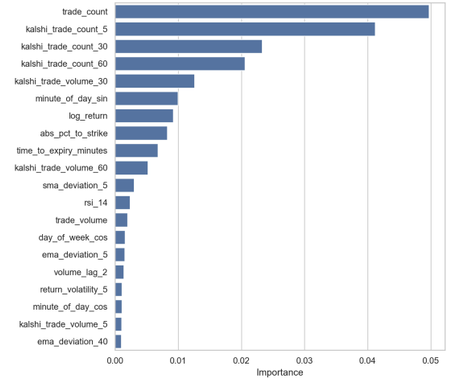
Eth Feature Comparison

APPENDIX FIGURE CATALOG

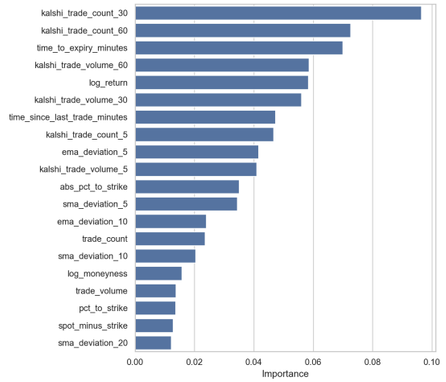
Generated figure backup



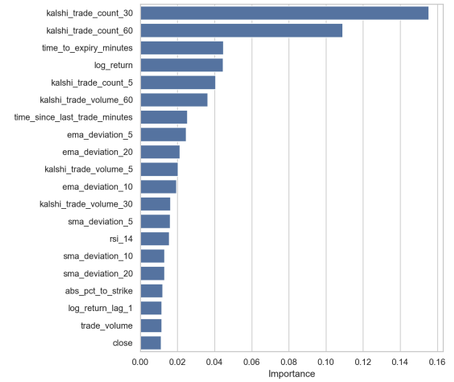
Feature Group Ablation



Feature Importance Mlp



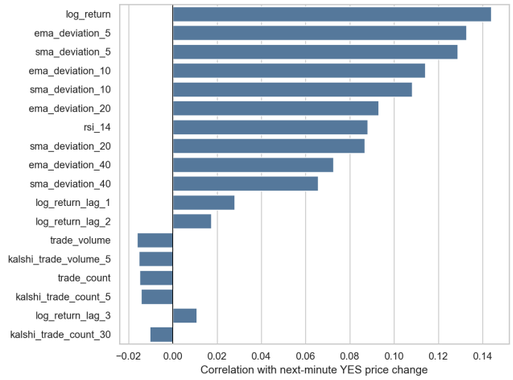
Feature Importance Random Forest



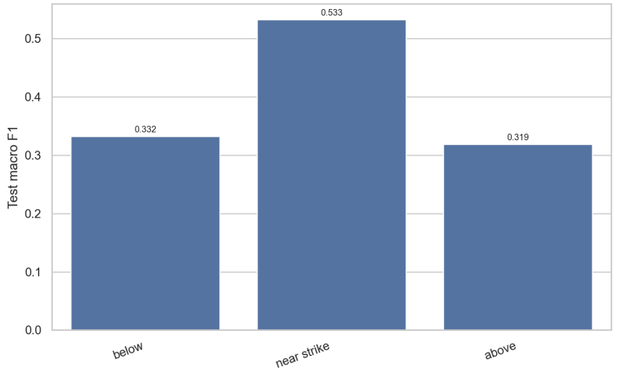
Feature Importance Xgboost

APPENDIX FIGURE CATALOG

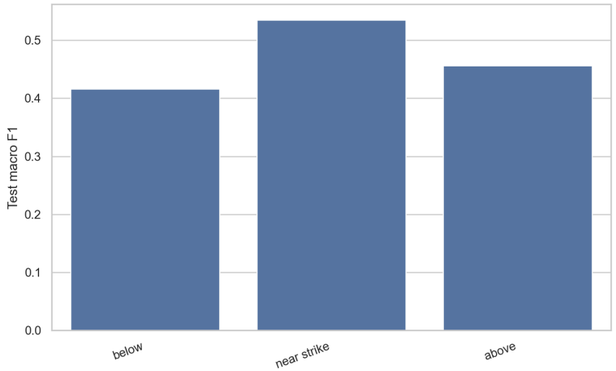
Generated figure backup



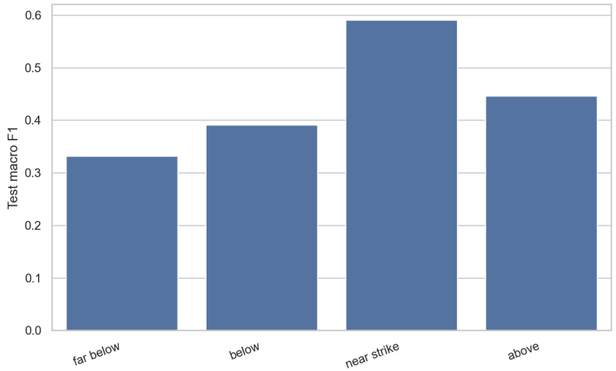
Feature Target Delta Correlation



Group Performance Moneyness Mlp



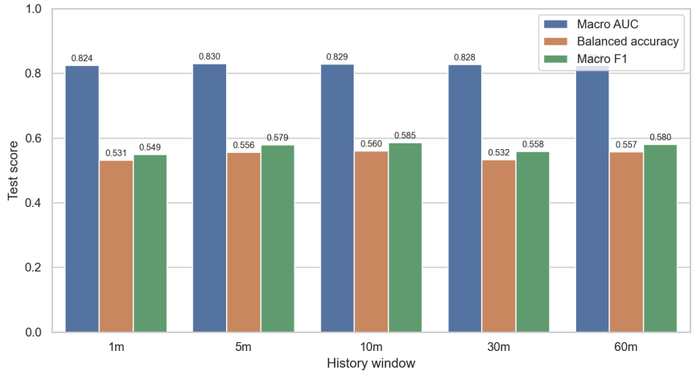
Group Performance Moneyness Random Forest



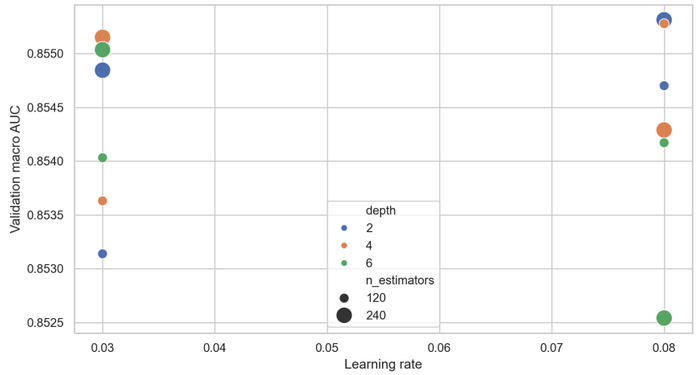
Group Performance Moneyness Xgboost

APPENDIX FIGURE CATALOG

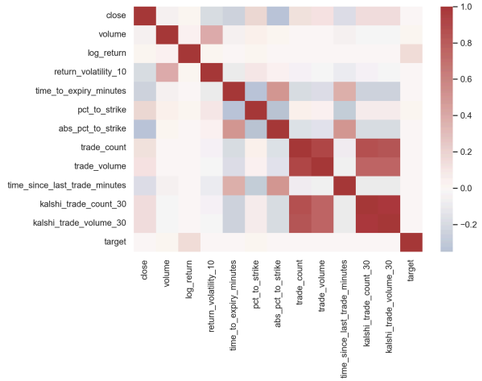
Generated figure backup



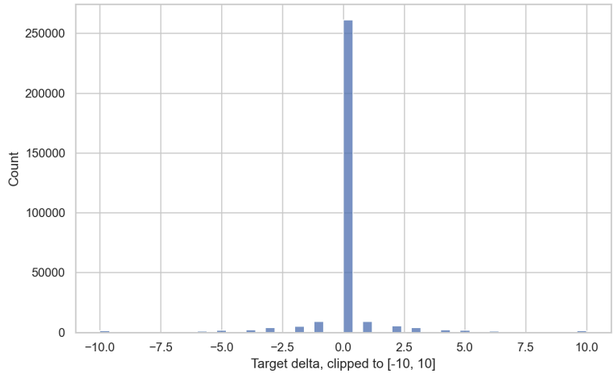
History Window Experiment



Hyperparameter Search



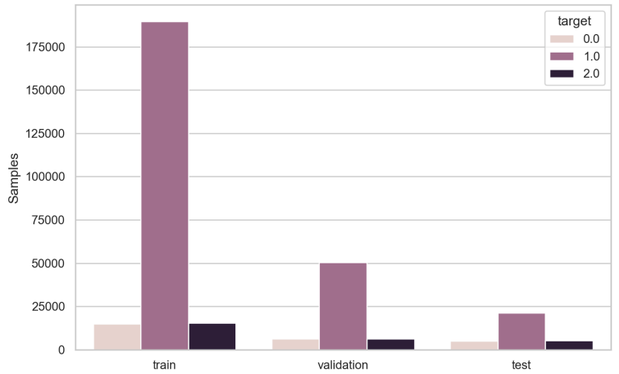
Inspection Feature Correlation Heatmap



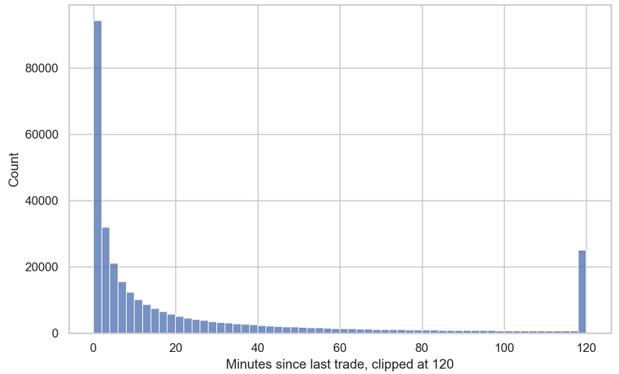
Inspection Target Delta Distribution

APPENDIX FIGURE CATALOG

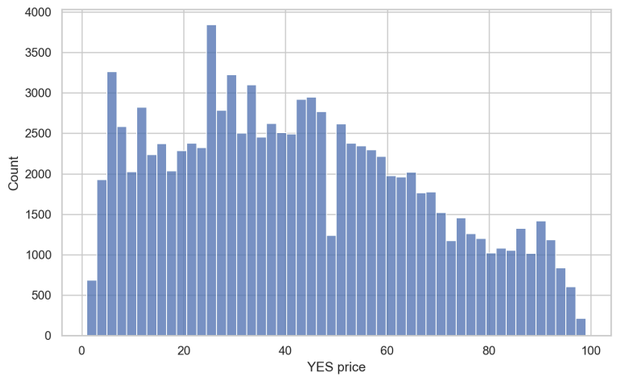
Generated figure backup



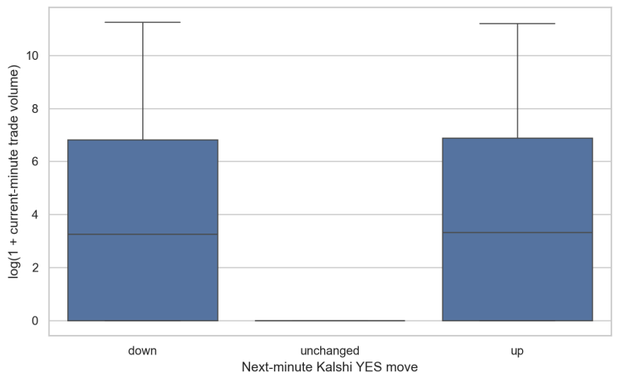
Inspection Target Distribution



Inspection Time Since Trade



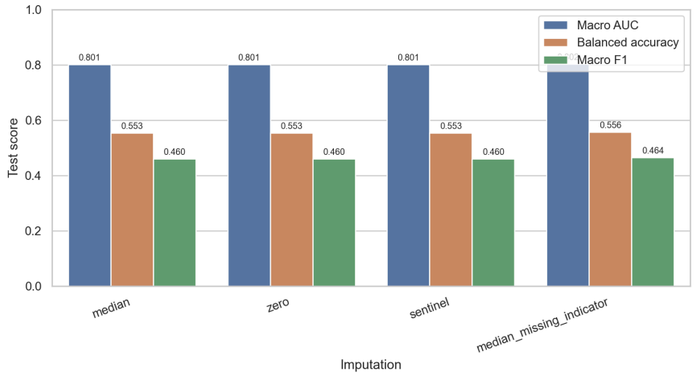
Inspection Yes Price Distribution



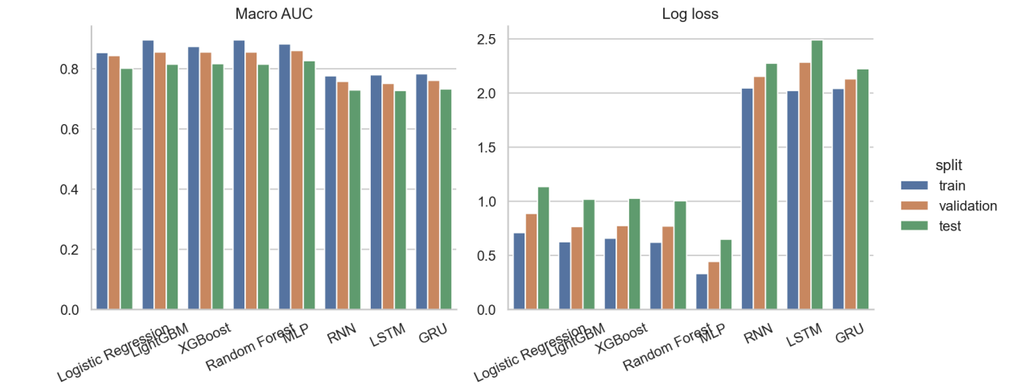
Kalshi Activity By Target

APPENDIX FIGURE CATALOG

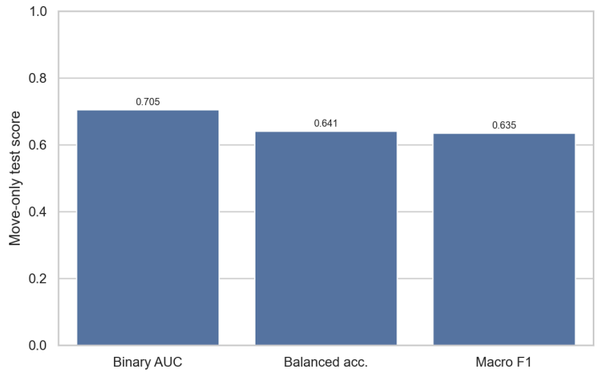
Generated figure backup



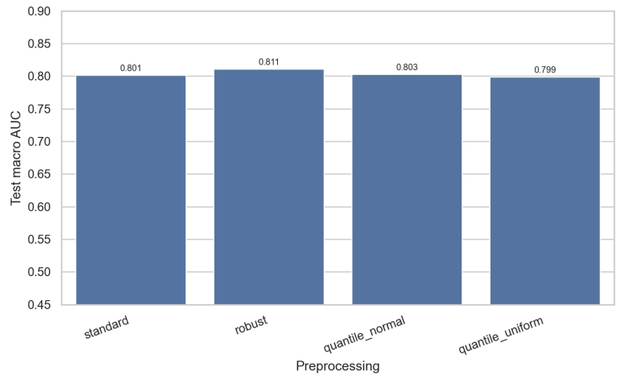
Missing Value Imputation Experiment



Model Auc Log Loss By Split



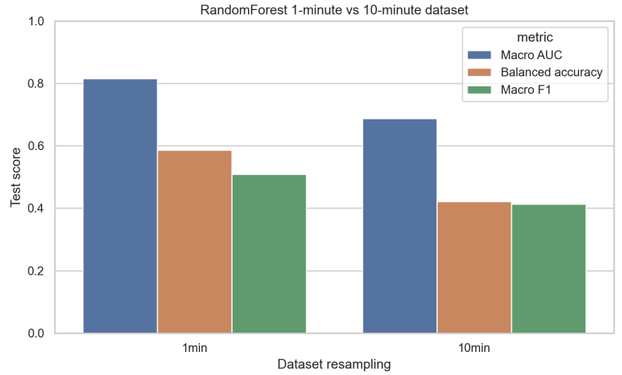
Move Direction Mlp



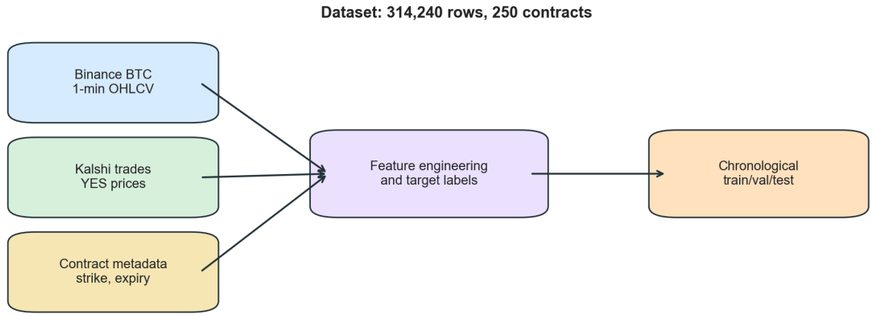
Preprocessing Experiment

APPENDIX FIGURE CATALOG

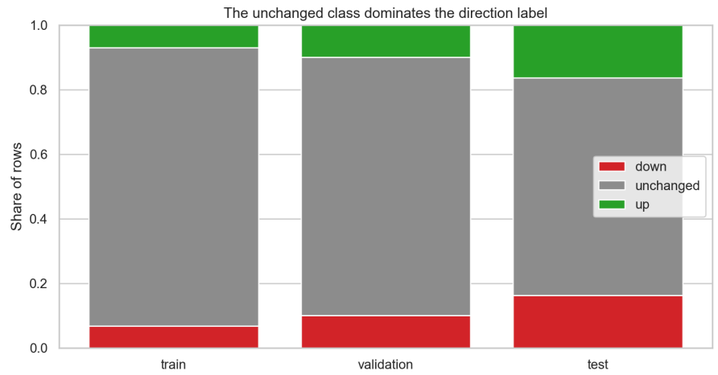
Generated figure backup



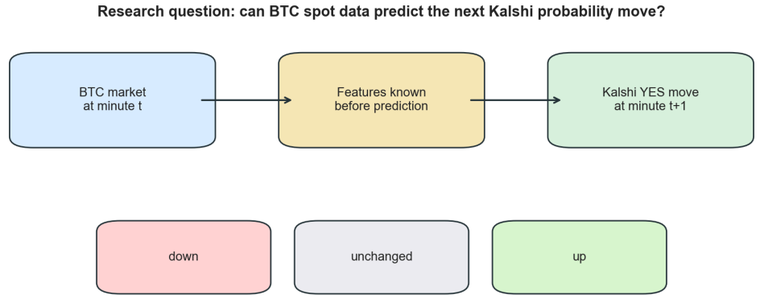
Random Forest 10M Comparison



Story Dataset Pipeline



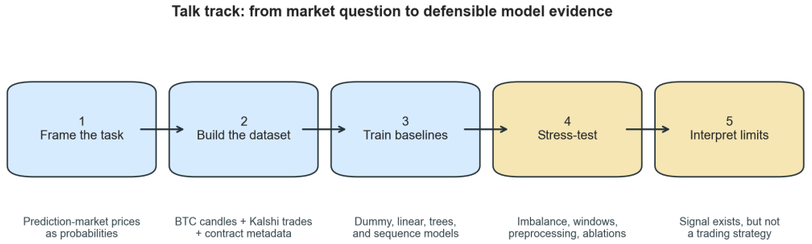
Story Label Imbalance



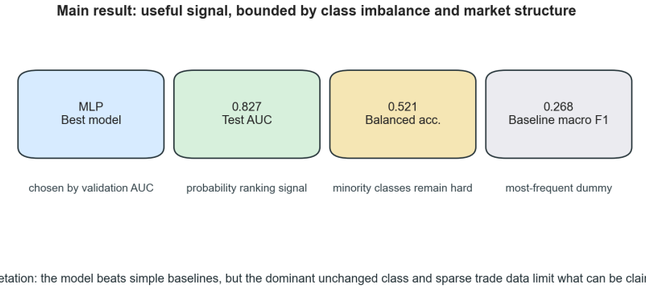
Story Problem Setup

APPENDIX FIGURE CATALOG

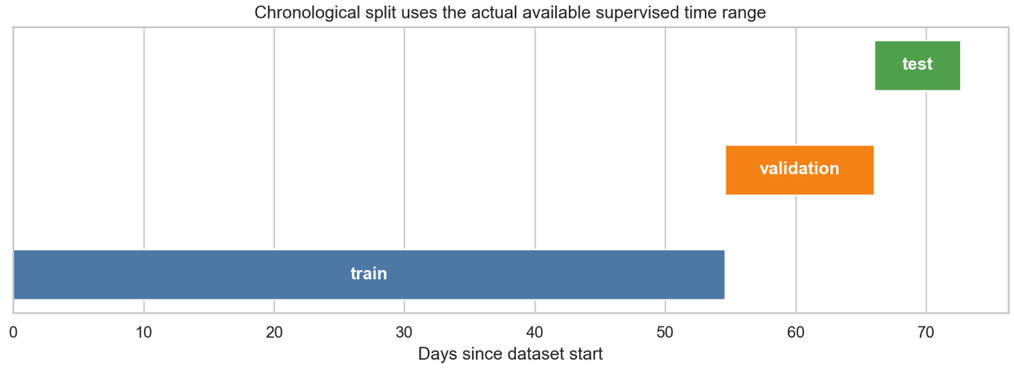
Generated figure backup



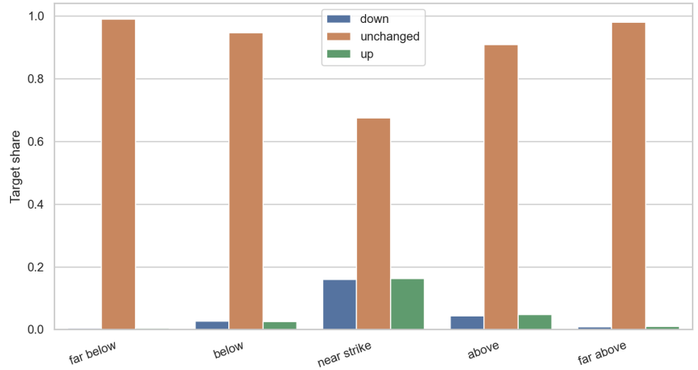
Story Research Roadmap



Story Result Takeaways

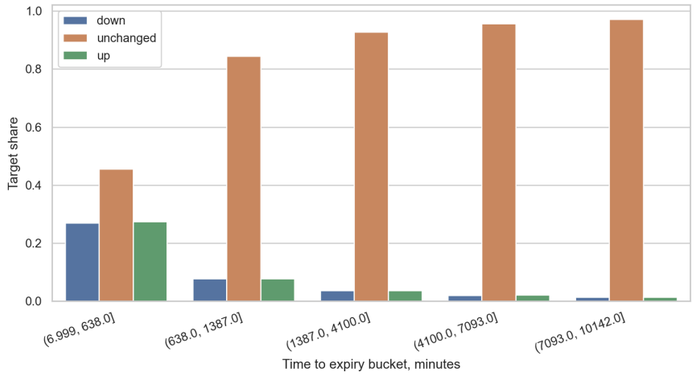


Story Split Timeline

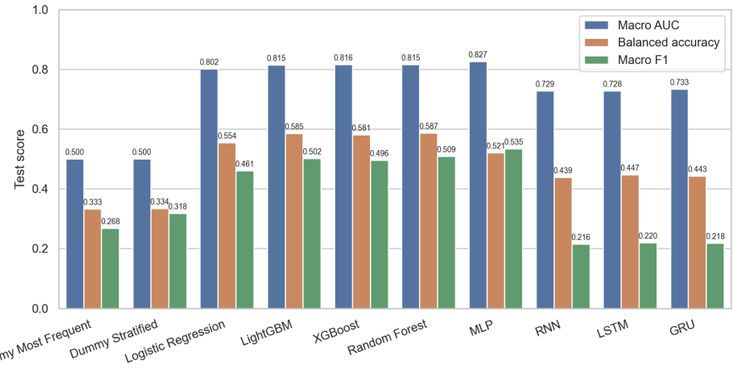


Target Mix By Moneyness

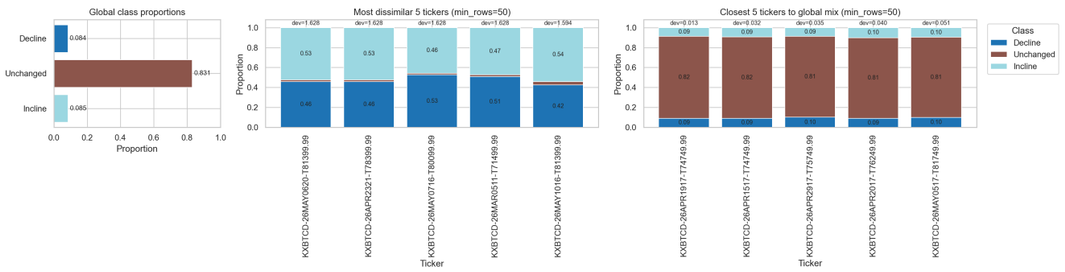
Generated figure backup



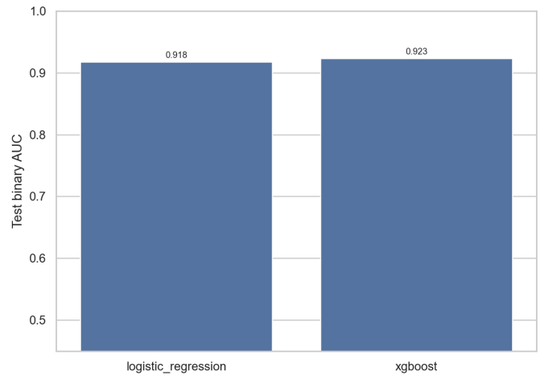
Target Mix By Time To Expiry



Test Model Comparison



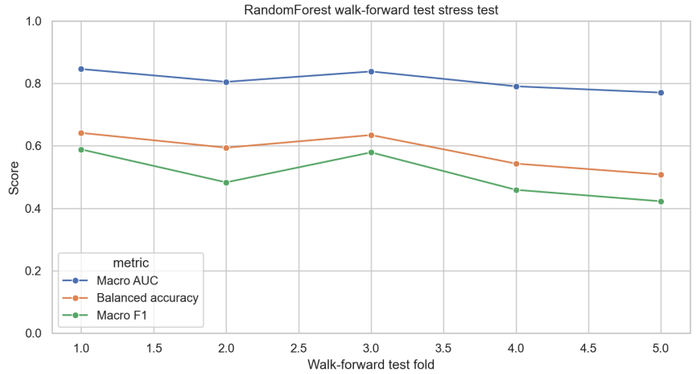
Ticker Class Distribution Deviation



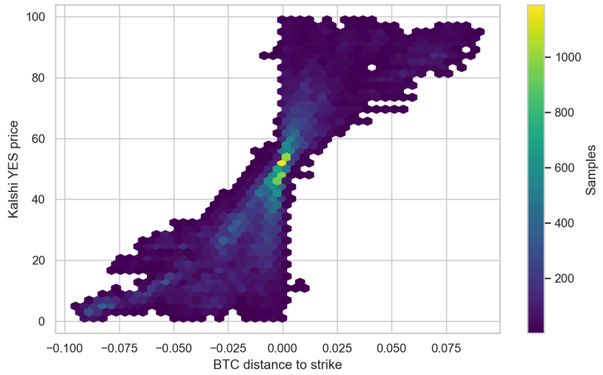
Volume Prediction

APPENDIX FIGURE CATALOG

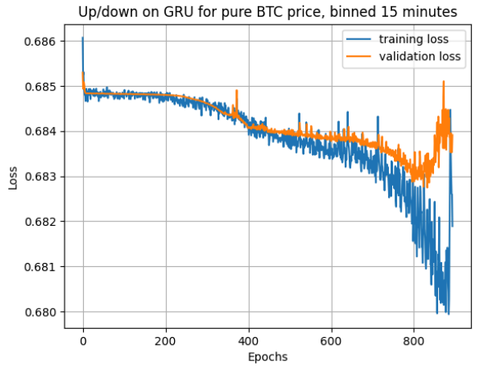
Generated figure backup



Walk Forward Test Cv



Yes Price Moneyness Density



Raw BTC Price Prediction