# MSFT Stock Market Trading Strategy via Classification

Zachary Mor

Brown University

https://github.com/zachmor/data1030-proj

December 5, 2022

# Recap

Predicting stock market using classification

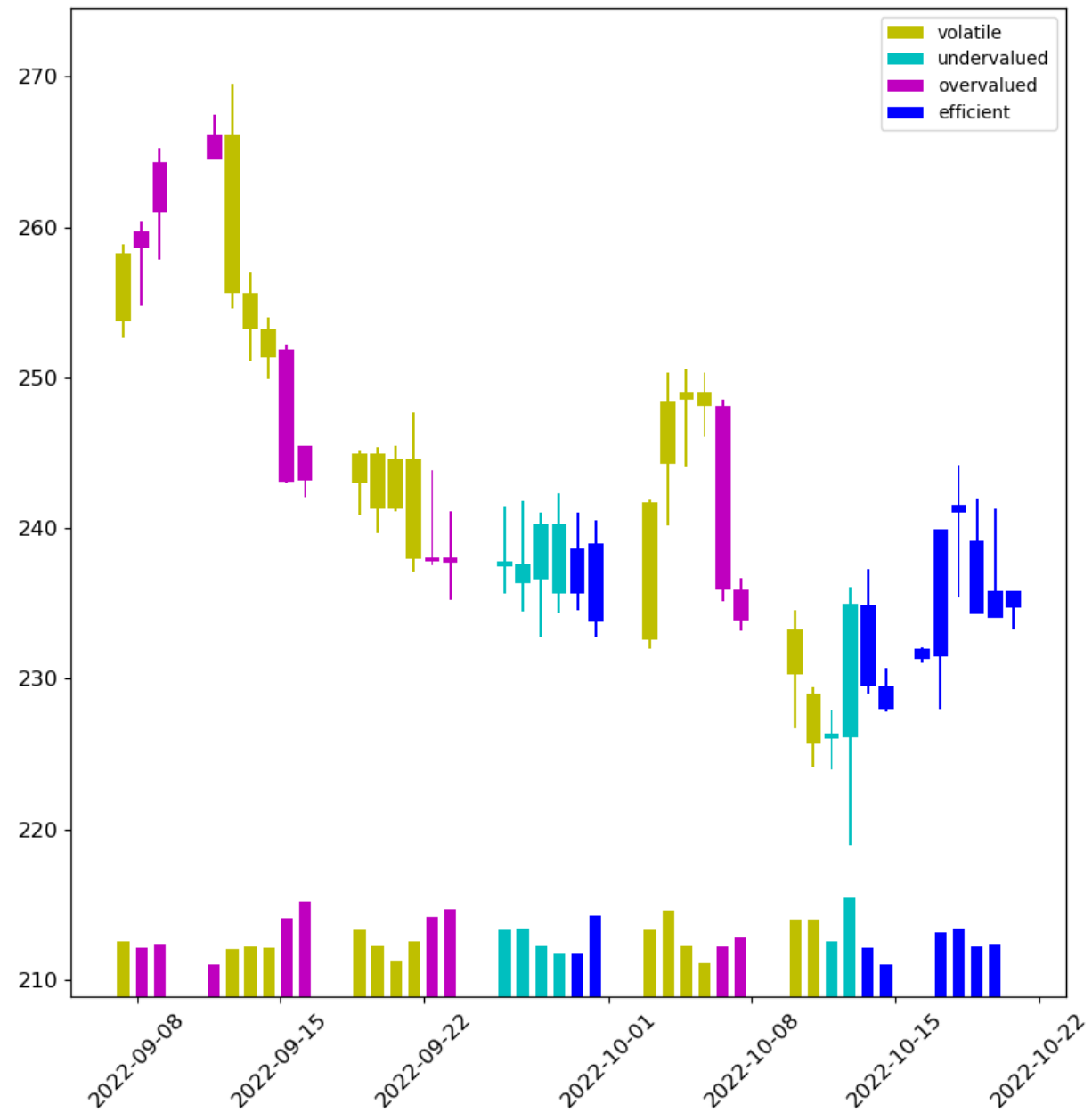Defined *inefficiencies* to consider weekly options

*i.e.* undervalued, overvalued, efficient, volatile
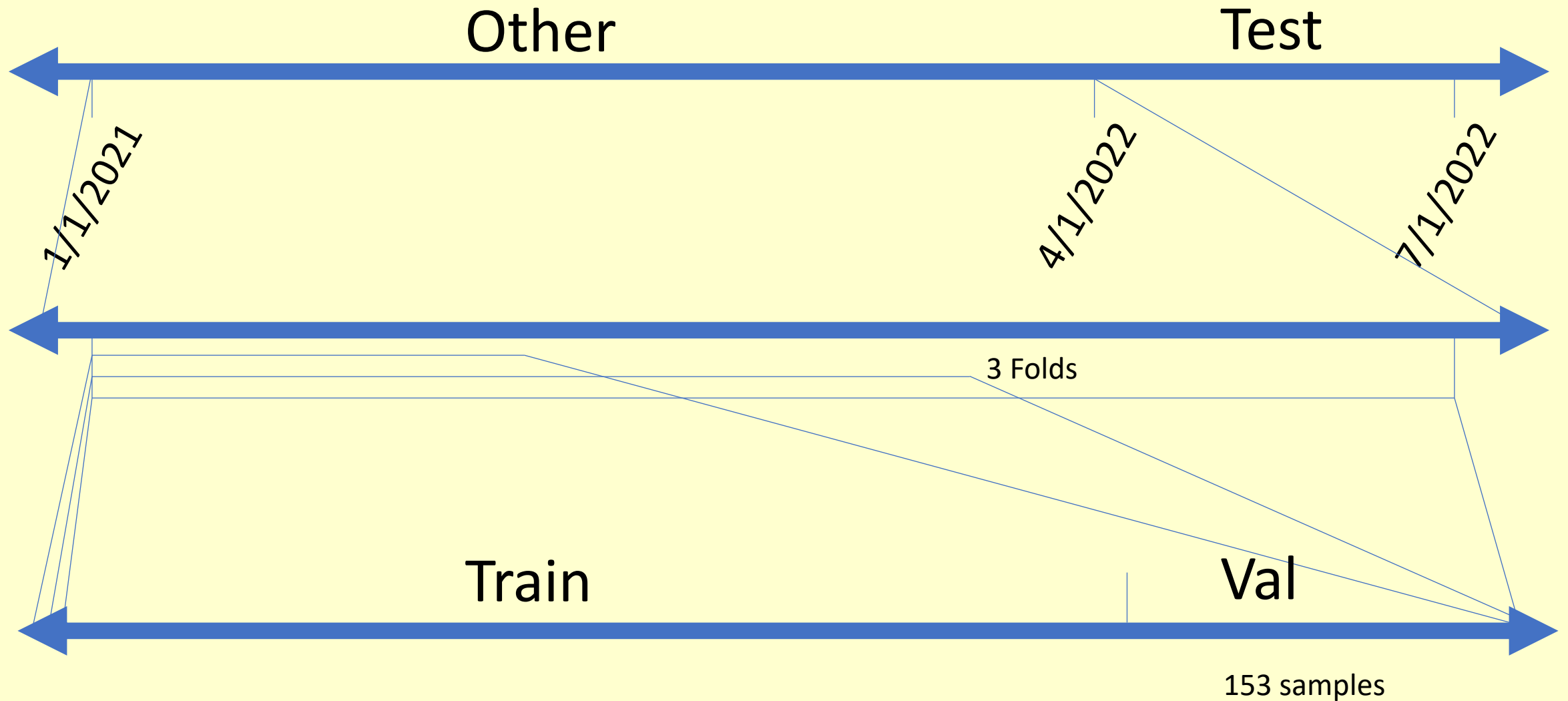
Inefficiencies(1 week, .02, 2 days)

On a given day, if *within 1 week,*

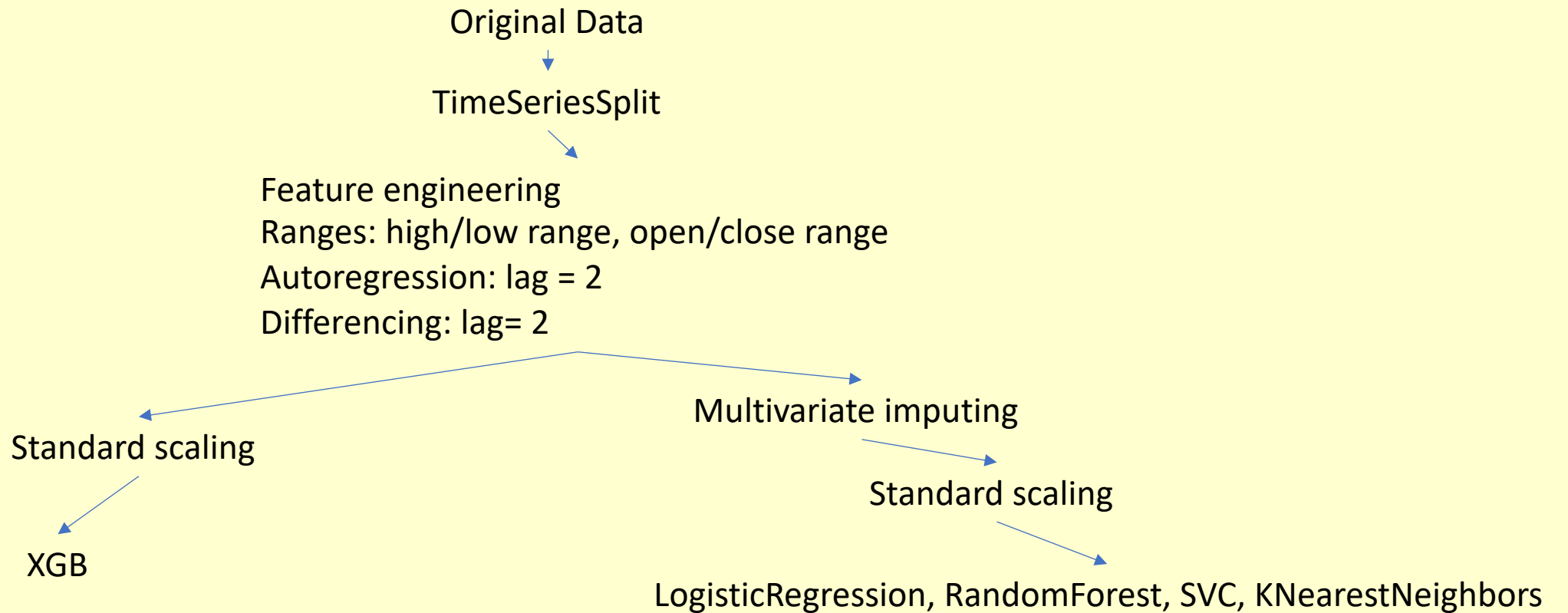the price goes up or down *at least 2%,*

And stays there for *at least 2 days.*

ALPHA
VANTAGE

MSFT Stock Inefficiencies,
delta > 2% by the end of the following work week for > 2 days

Cross Validation: TimeSeriesSplit

Other

Test

1/1/2021

4/1/2022

7/1/2022

3 Folds

Train

Val

153 samples

# Pipeline

Original Data

TimeSeriesSplit

Feature engineering
Ranges: high/low range, open/close range
Autoregression: lag = 2
Differencing: lag= 2

Multivariate imputing

Standard scaling

Standard scaling

XGB

LogisticRegression, RandomForest, SVC, KNearestNeighbors

# Parameters

```python
{'randomforestclassifier__n_estimators': [100, 200, 300],
  'randomforestclassifier__max_depth': [3, 5, 7, 9],
  'randomforestclassifier__max_features': [0.1,0.3,0.5,0.7]},


 {'svc__C': np.logspace(-1,3,5),
  'svc__gamma': np.logspace(-1,2,4)},


{'kneighborsclassifier__n_neighbors': [1,3,10,30,100],
  'kneighborsclassifier__p': [1,2],
  'kneighborsclassifier__weights': ['uniform', 'distance']},


{'logisticregression__penalty' : ['elasticnet'],
 'logisticregression__solver' : ['saga'],
 'logisticregression__l1_ratio' : np.linspace(.1,.9, 5),
 'logisticregression__C' : np.logspace(-3,3,7),
 'logisticregression__max_iter' : [1000000]}
```

```python
{"xgbclassifier__learning_rate": [0.03],
     "xgbclassifier__n_estimators": [30,100,300,1000,3000],
     "xgbclassifier__seed": [0],
     "xgbclassifier__reg_alpha": [1e-2, 1e-1],
     "xgbclassifier__reg_lambda": [1e-2, 1e-1],
     "xgbclassifier__missing": [np.nan],
     "xgbclassifier__max_depth": [3,10,30],
     # "xgbclassifier__colsample_bytree": [0.9],
     "xgbclassifier__subsample": [0.66]}
```
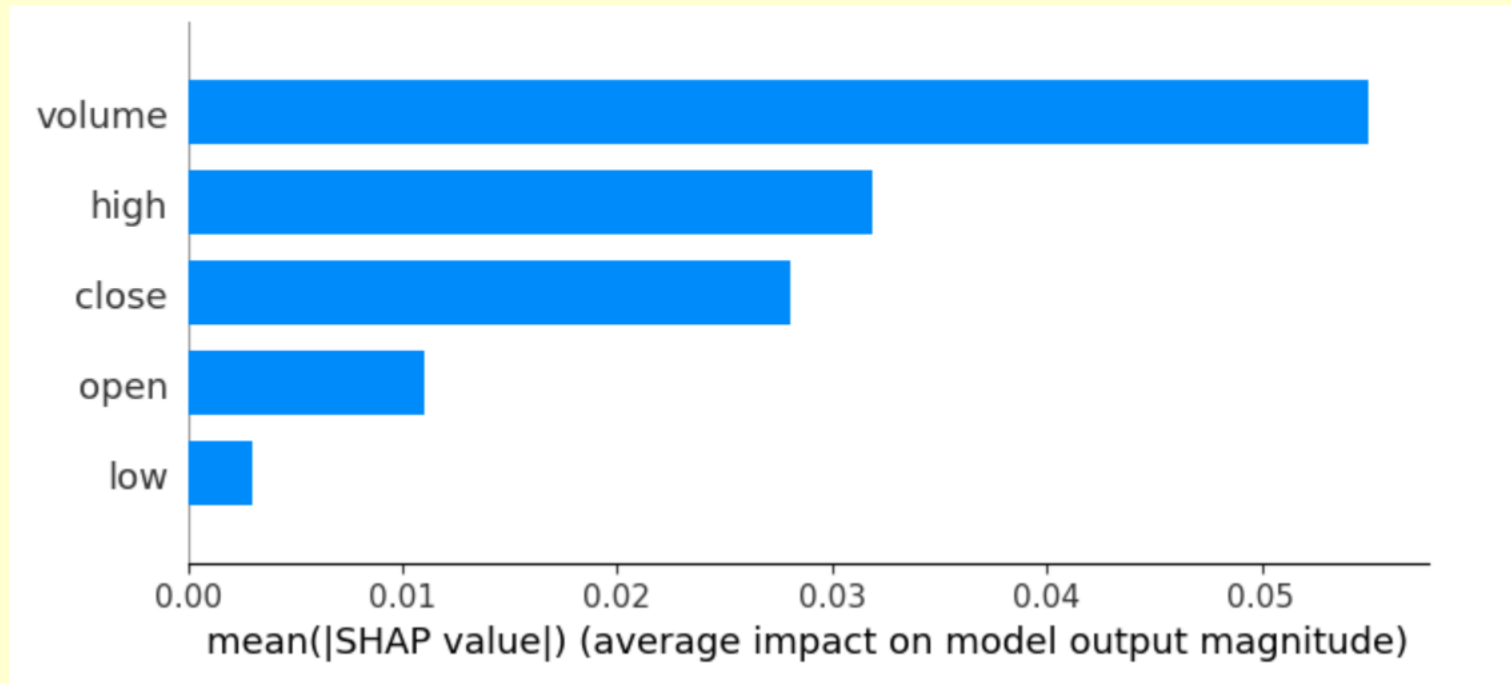
35 min runtime :D

# Results

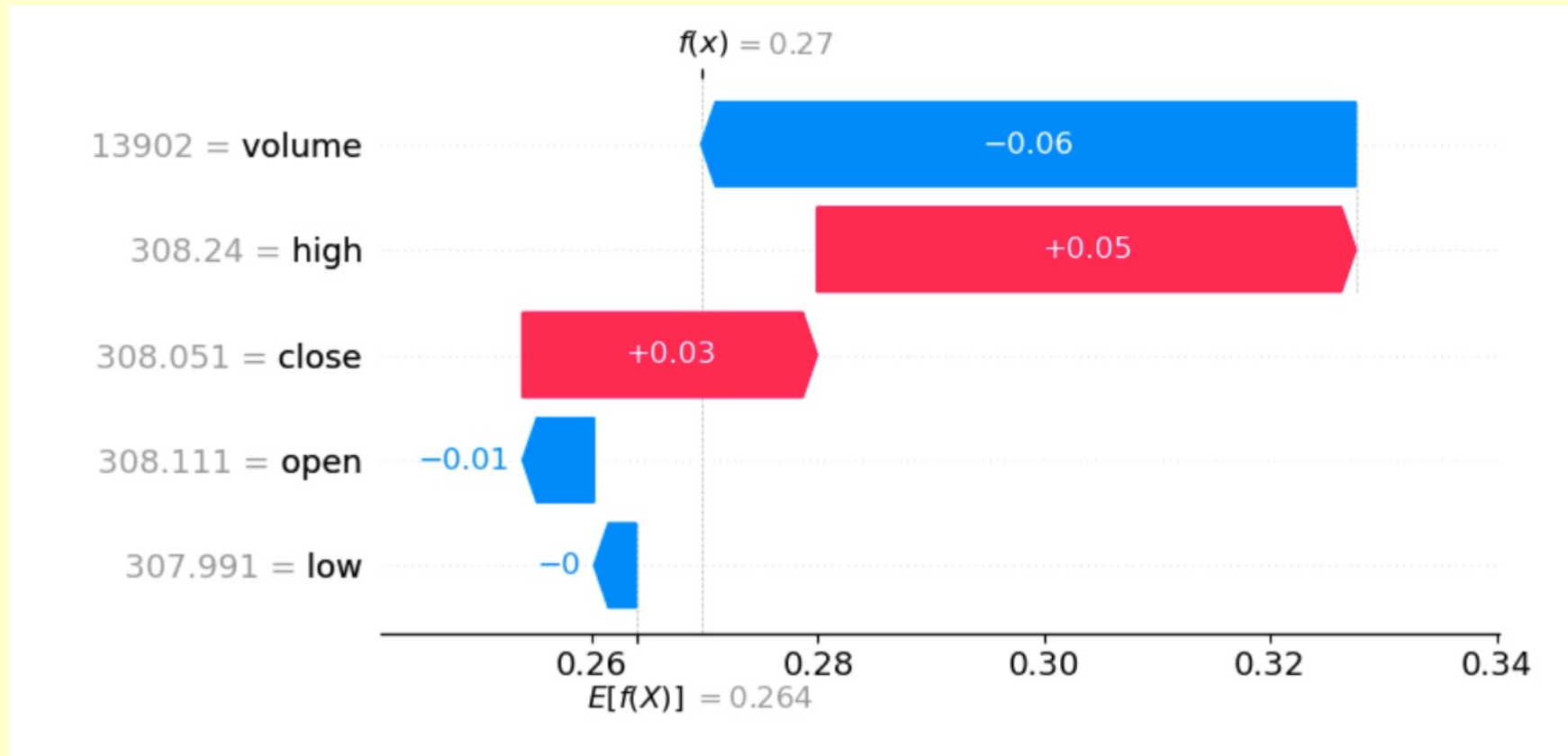baseline from X_other,
always underpriced
only 13% on test.

baseline from X_test,
always overpriced,
44%.

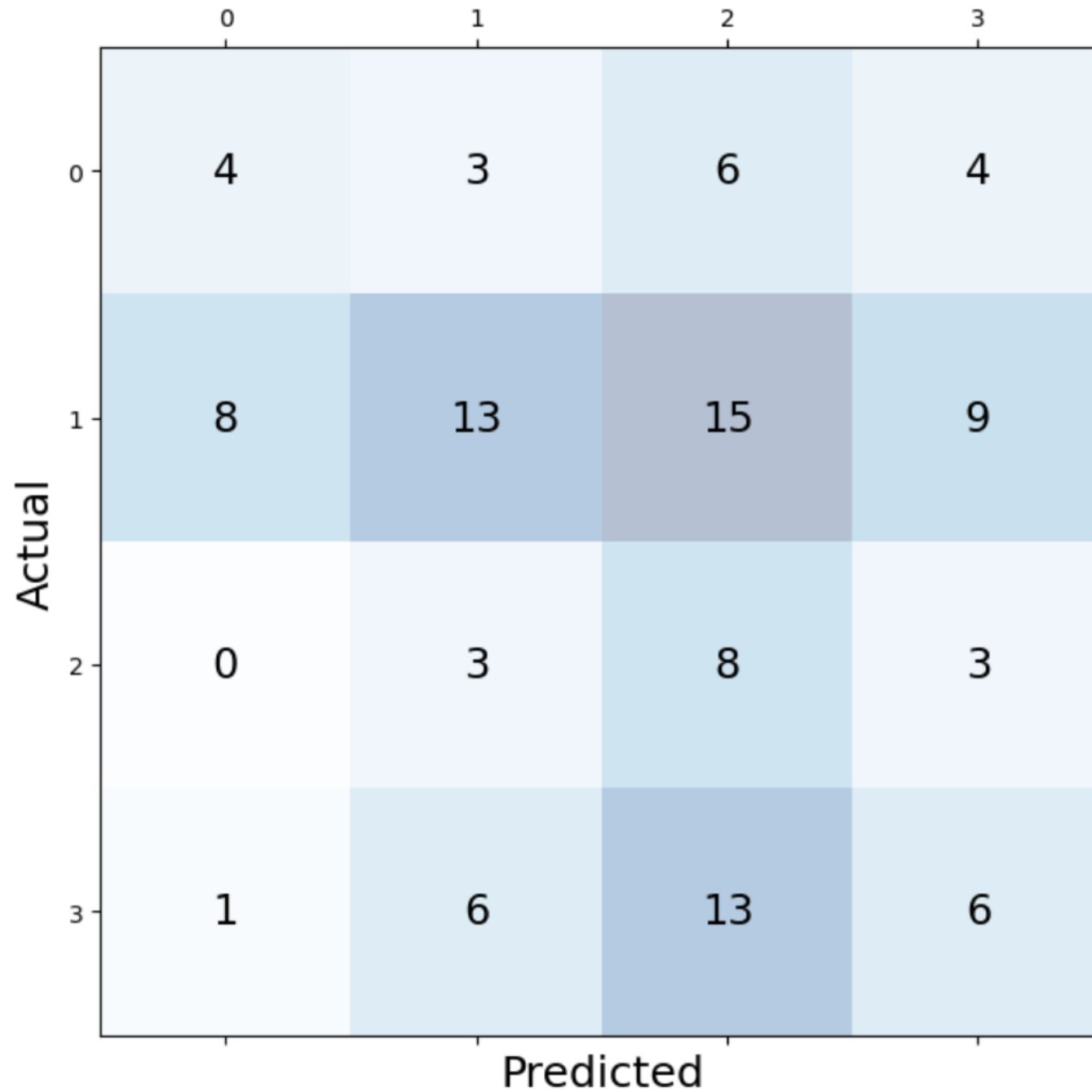| name | test score |
|---|---|
| kneighborsclassifier | 0.303922 |
| logisticregression | 0.137255 |
| randomforestclassifier | 0.225490 |
| svc | 0.215686 |
| xgboostclassifier | 0.196078 |

# Global Feature Importance

# Local Feature Importance

Net Confusion Matrix

# Outlooks

- [ORIGINAL PROJECT] The rest of the data…
- Multiple frequencies
- Higher resolution interpretability if I preprocess separately,
  - Tricky with sliding windows and CV

# Questions