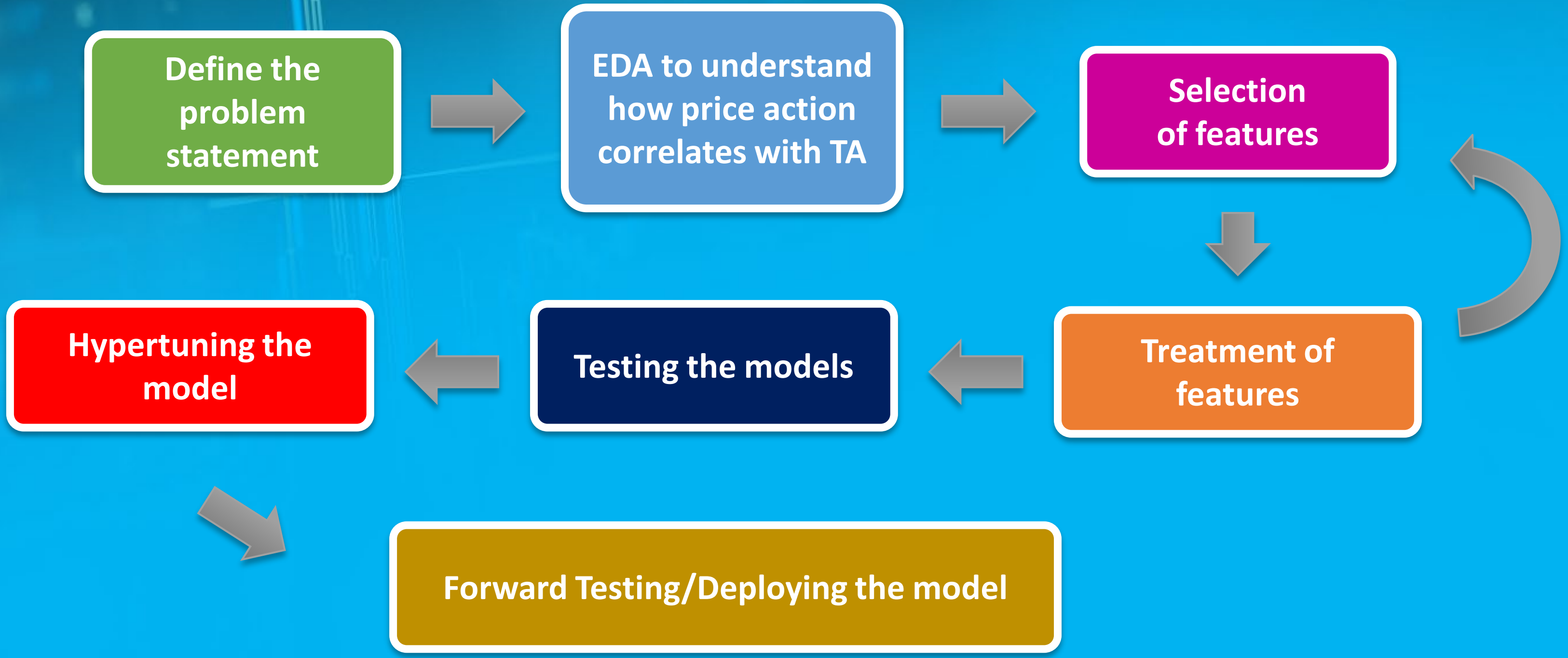
DS105 Project
Predicting Stock Prices

Edwin Wan

How the ML process was executed?



Define the problem statement

EDA to understand how price action correlates with TA

Selection of features

Treatment of features

Testing the models

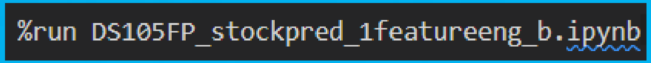
Hypertuning the model

Forward Testing/Deploying the model

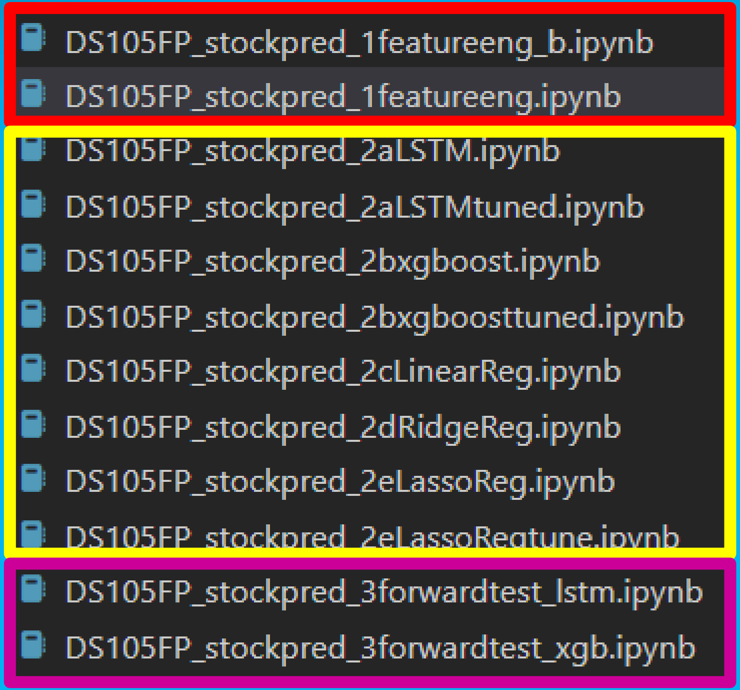
How the ML process was executed?

- Each step of the ML was split into individual notebooks

- Make it more efficient when adjustments need to be made



```
%run DS105FP_stockpred_1featureeng_b.ipynb
```



DS105FP_stockpred_1featureeng_b.ipynb

DS105FP_stockpred_1featureeng.ipynb

DS105FP_stockpred_2aLSTM.ipynb

DS105FP_stockpred_2aLSTMtuned.ipynb

DS105FP_stockpred_2bxgboost.ipynb

DS105FP_stockpred_2bxgboosttuned.ipynb

DS105FP_stockpred_2cLinearReg.ipynb

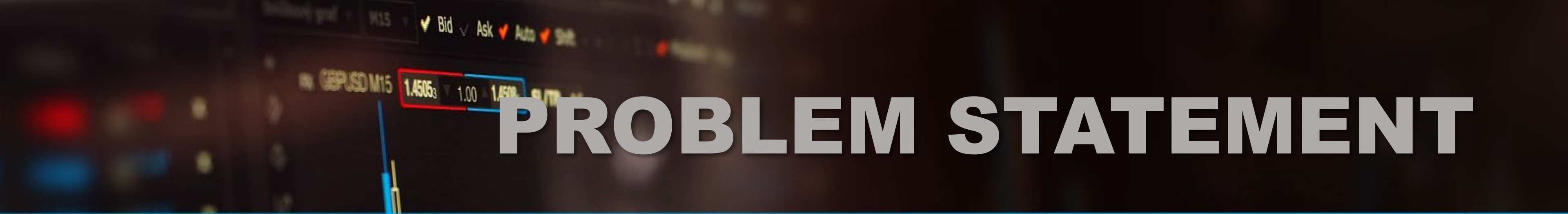
DS105FP_stockpred_2dRidgeReg.ipynb

DS105FP_stockpred_2eLassoReg.ipynb

DS105FP_stockpred_2eLassoRegtune.ipynb

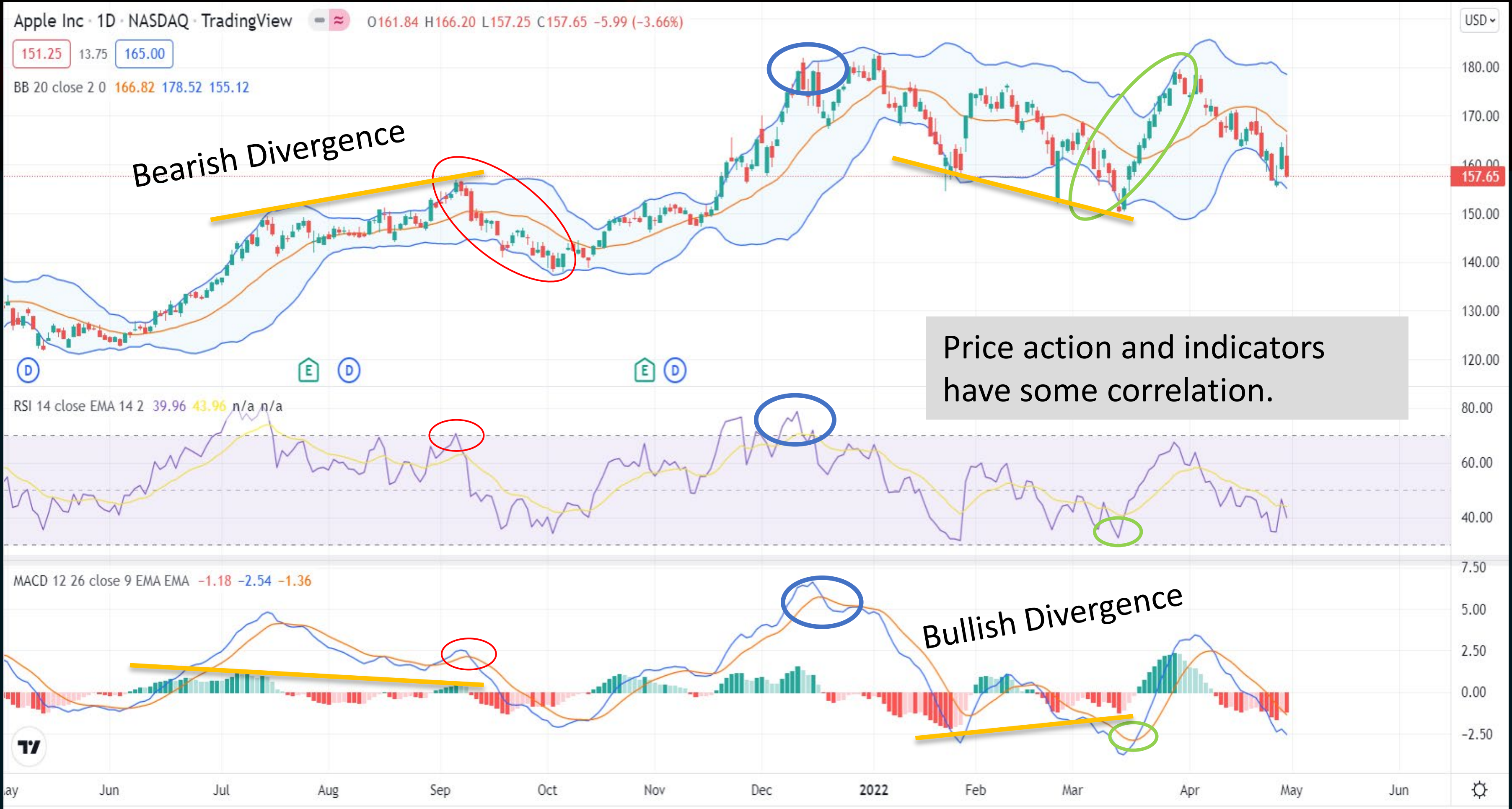
DS105FP_stockpred_3forwardtest_lstm.ipynb

DS105FP_stockpred_3forwardtest_xgb.ipynb



PROBLEM STATEMENT

Forecast stock prices up to 3 days ahead to see how forecasted prices will interact with resistance/support to decide on options strategy.



Bearish Divergence

Bullish Divergence

Price action and indicators have some correlation.

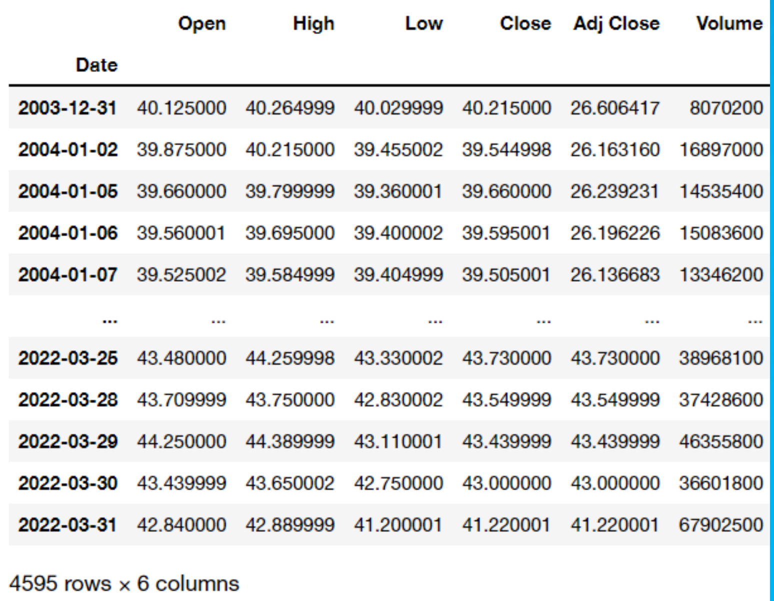
Deciding on the features

- Deciding on the price lag (no of lag days for adjusted close price) – Up to Day-3

- Deciding which **initial** indicator
  - Bband, MACD, RSI, MFI

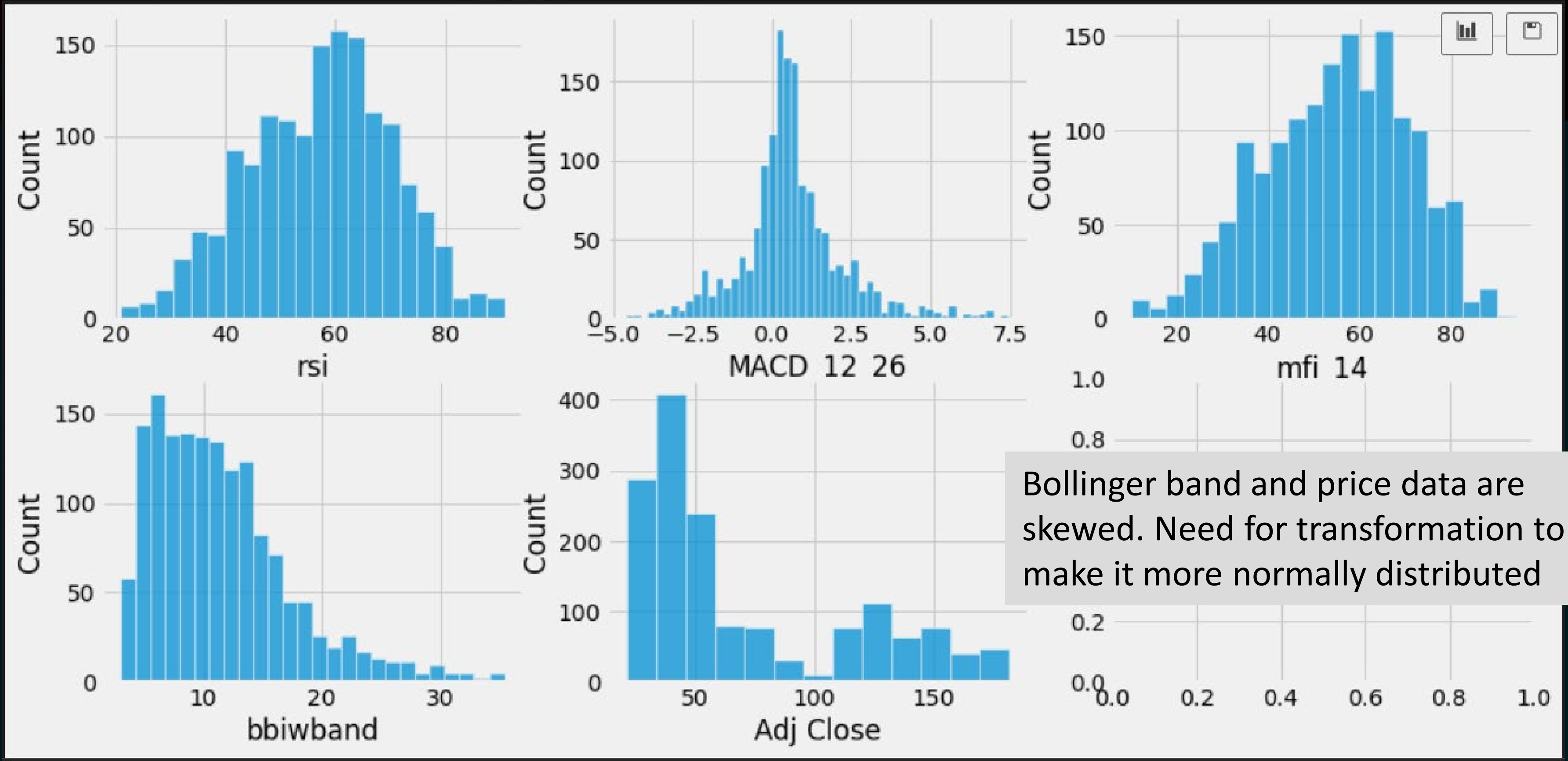
Datasets

- Price data from Yahoo Finance API
- Focus on **<Adj Close>** price data
- Use data from **2016 to 28th April 2022** instead as the growth rate was more consistent to current date
- Features will be engineered using the adj close price



	Open	High	Low	Close	Adj Close	Volume
Date						
2003-12-31	40.125000	40.264999	40.029999	40.215000	26.606417	8070200
2004-01-02	39.875000	40.215000	39.455002	39.544998	26.163160	16897000
2004-01-05	39.660000	39.799999	39.360001	39.660000	26.239231	14535400
2004-01-06	39.560001	39.695000	39.400002	39.595001	26.196226	15083600
2004-01-07	39.525002	39.584999	39.404999	39.505001	26.136683	13346200
...	...	...	...	...	...	...
2022-03-25	43.480000	44.259998	43.330002	43.730000	43.730000	38968100
2022-03-28	43.709999	43.750000	42.830002	43.549999	43.549999	37428600
2022-03-29	44.250000	44.389999	43.110001	43.439999	43.439999	46355800
2022-03-30	43.439999	43.650002	42.750000	43.000000	43.000000	36601800
2022-03-31	42.840000	42.889999	41.200001	41.220001	41.220001	67902500

4595 rows × 6 columns



Bollinger band and price data are skewed. Need for transformation to make it more normally distributed

Deciding on the features

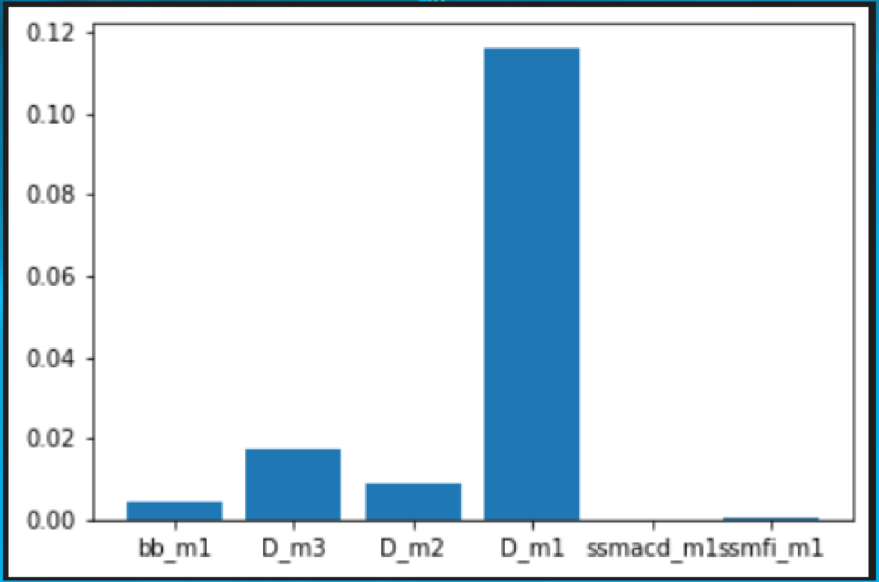
- Train-Val-Test Split
  - Train: 85%
  - Val: 7%
  - Test: 8%

- Treatments of Data
  - Base Model(Raw data)
  - Binning of RSI and MFI data
  - Scaling MACD
  - Log Transformed/Pct change of price and Bband data

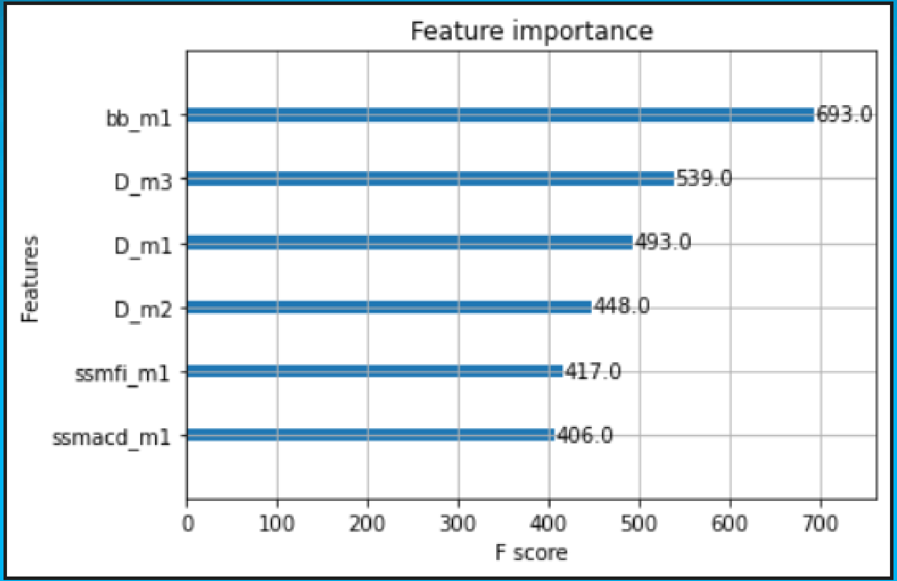
type1	basemodel + binning of MFI
type2	basemodel + scaling and log trasformation of price values
type3	basemodel + scaling and pct_chg of price values
type4	type 1 + type 2
type5	type 1 + type 3

Processing type	**RMSE**	**MAPE**
base model	1.036	0.03
type1	1.431	0.031
type2	6.674	0.058
type3	0.901	0.0296
type5	1.061	0.0298

IMPORTANCE OF FEATURES



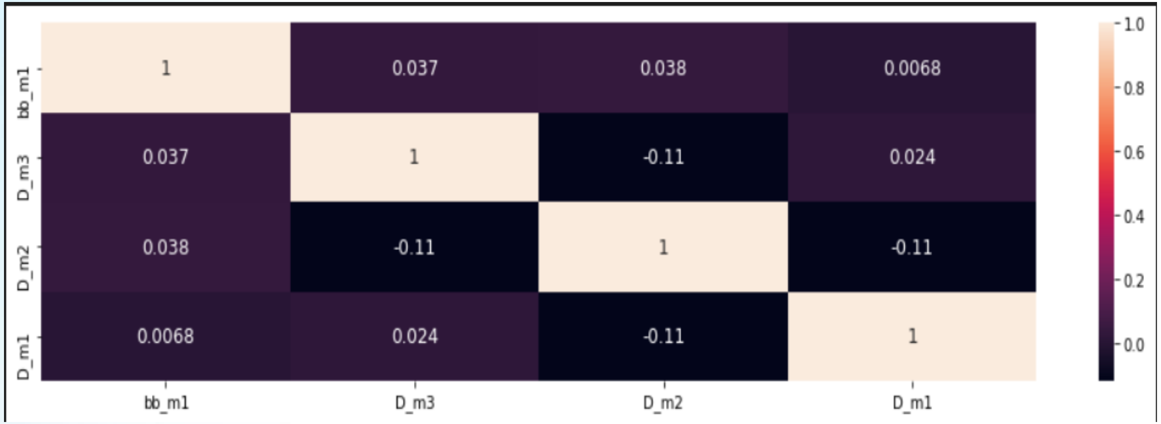
Coefficient values taken from linear regression model



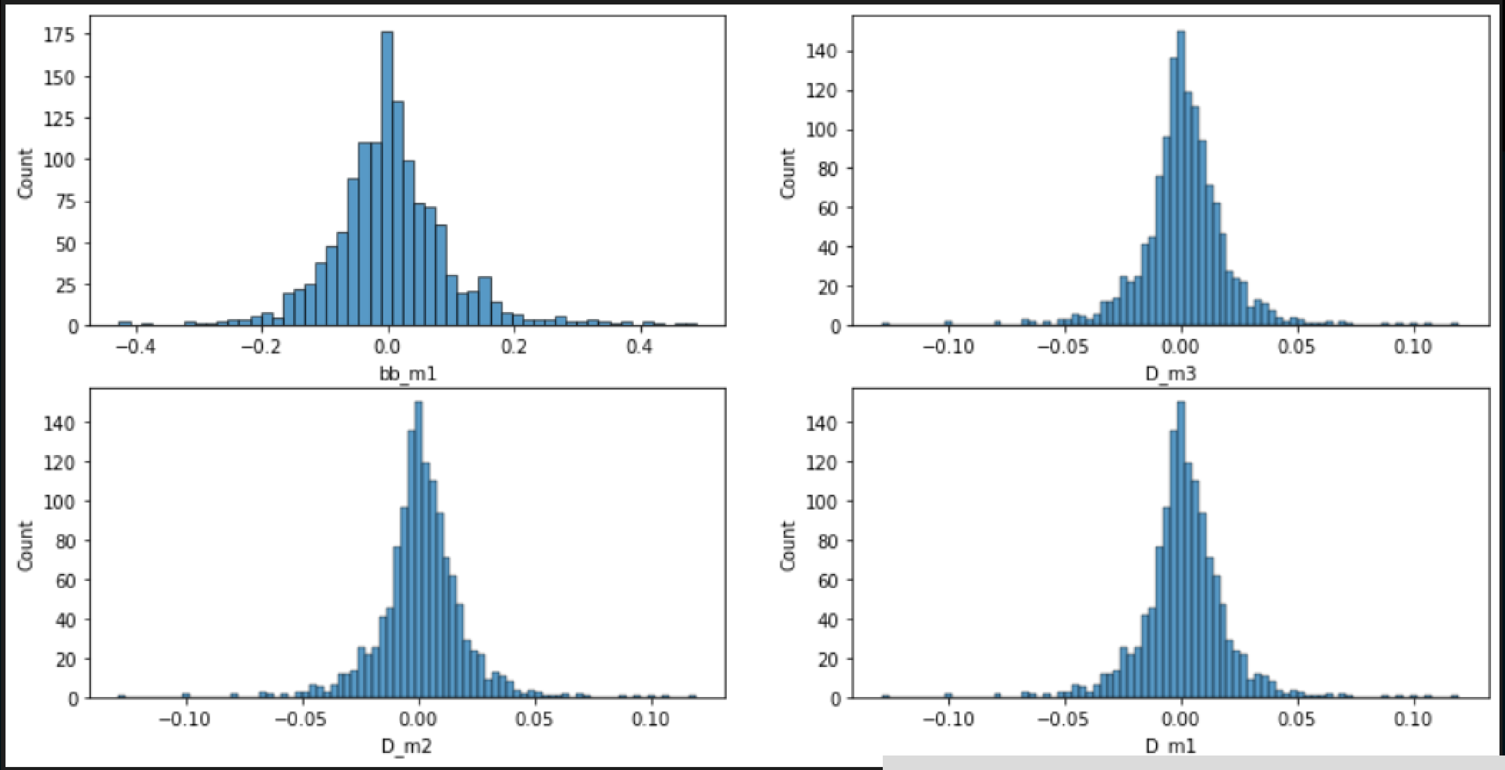
Feature importance from XGBoost model

Final selected features

- Previous 3 day adjusted close price
- Bollinger band (1 day previous)
- Transformed with percent change



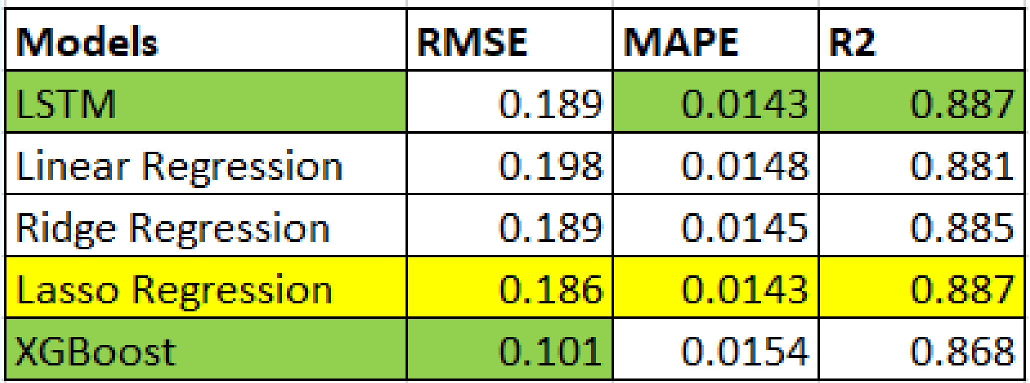
	bb_m1	D_m3	D_m2	D_m1
bb_m1	1	0.037	0.038	0.0068
D_m3	0.037	1	-0.11	0.024
D_m2	0.038	-0.11	1	-0.11
D_m1	0.0068	0.024	-0.11	1



With percent change transformation, data is more normally distributed

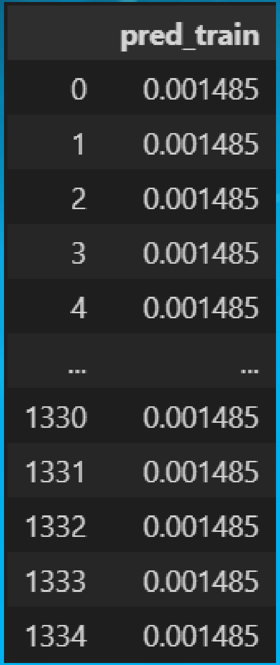
ML Models tested

1. Liner Regression
2. Ridge Regression
3. Lasso Regression
4. LSTM (Deep Learning)
5. XGBoost

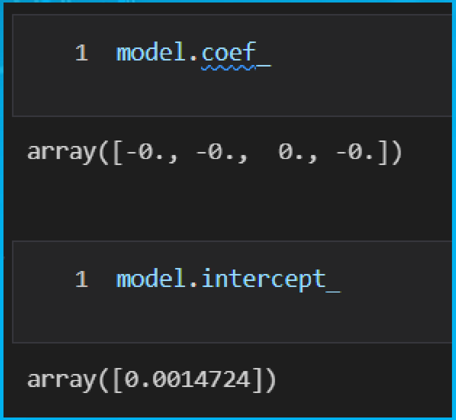


Models	RMSE	MAPE	R2
LSTM	0.189	0.0143	0.887
Linear Regression	0.198	0.0148	0.881
Ridge Regression	0.189	0.0145	0.885
Lasso Regression	0.186	0.0143	0.887
XGBoost	0.101	0.0154	0.868

Why Lasso Regression was dropped?



	pred_train
0	0.001485
1	0.001485
2	0.001485
3	0.001485
4	0.001485
...	...
1330	0.001485
1331	0.001485
1332	0.001485
1333	0.001485
1334	0.001485



```
model.coef_
```

```
array([-0., -0.,  0., -0.])
```

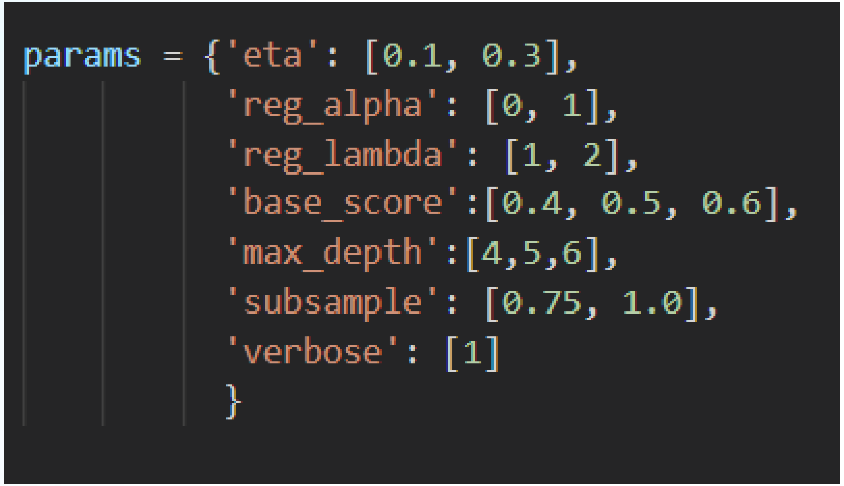
```
model.intercept_
```

```
array([0.0014724])
```

- Realised predicted pct change was the same for all features
- Model Coefficient was 0
- Predicted values were just using intercept values

Hyper Parameter Tuning

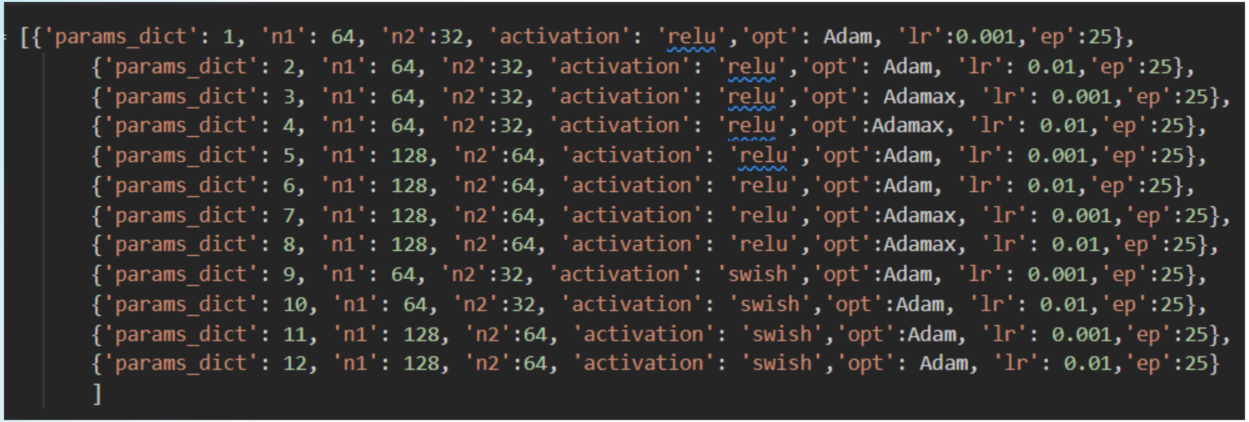
- Gridsearch CV was used for XGBoost Model



```
params = {'eta': [0.1, 0.3],
          'reg_alpha': [0, 1],
          'reg_lambda': [1, 2],
          'base_score':[0.4, 0.5, 0.6],
          'max_depth':[4,5,6],
          'subsample': [0.75, 1.0],
          'verbose': [1]
          }
```

Hyper Parameter Tuning

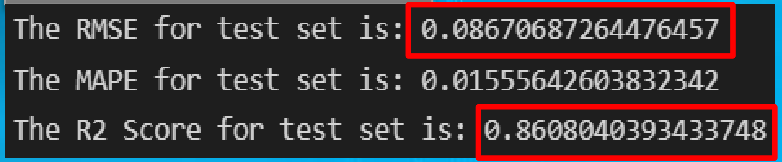
- For Loop used for tuning of LSTM



```
[{'params_dict': 1, 'n1': 64, 'n2':32, 'activation': 'relu','opt': Adam, 'lr':0.001,'ep':25},
      {'params_dict': 2, 'n1': 64, 'n2':32, 'activation': 'relu','opt': Adam, 'lr': 0.01,'ep':25},
      {'params_dict': 3, 'n1': 64, 'n2':32, 'activation': 'relu','opt': Adamax, 'lr': 0.001,'ep':25},
      {'params_dict': 4, 'n1': 64, 'n2':32, 'activation': 'relu','opt':Adamax, 'lr': 0.01,'ep':25},
      {'params_dict': 5, 'n1': 128, 'n2':64, 'activation': 'relu','opt':Adam, 'lr': 0.001,'ep':25},
      {'params_dict': 6, 'n1': 128, 'n2':64, 'activation': 'relu','opt':Adam, 'lr': 0.01,'ep':25},
      {'params_dict': 7, 'n1': 128, 'n2':64, 'activation': 'relu','opt':Adamax, 'lr': 0.001,'ep':25},
      {'params_dict': 8, 'n1': 128, 'n2':64, 'activation': 'relu','opt':Adamax, 'lr': 0.01,'ep':25},
      {'params_dict': 9, 'n1': 64, 'n2':32, 'activation': 'swish','opt':Adam, 'lr': 0.001,'ep':25},
      {'params_dict': 10, 'n1': 64, 'n2':32, 'activation': 'swish','opt':Adam, 'lr': 0.01,'ep':25},
      {'params_dict': 11, 'n1': 128, 'n2':64, 'activation': 'swish','opt':Adam, 'lr': 0.001,'ep':25},
      {'params_dict': 12, 'n1': 128, 'n2':64, 'activation': 'swish','opt': Adam, 'lr': 0.01,'ep':25}
      ]
```

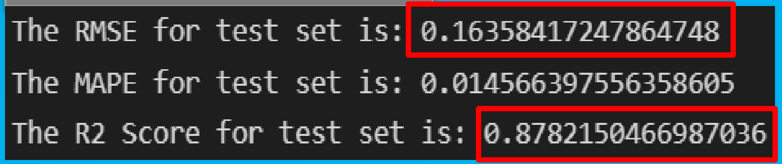
XGBoost Tuning

Base Model



```
The RMSE for test set is: 0.08670687264476457
The MAPE for test set is: 0.01555642603832342
The R2 Score for test set is: 0.8608040393433748
```

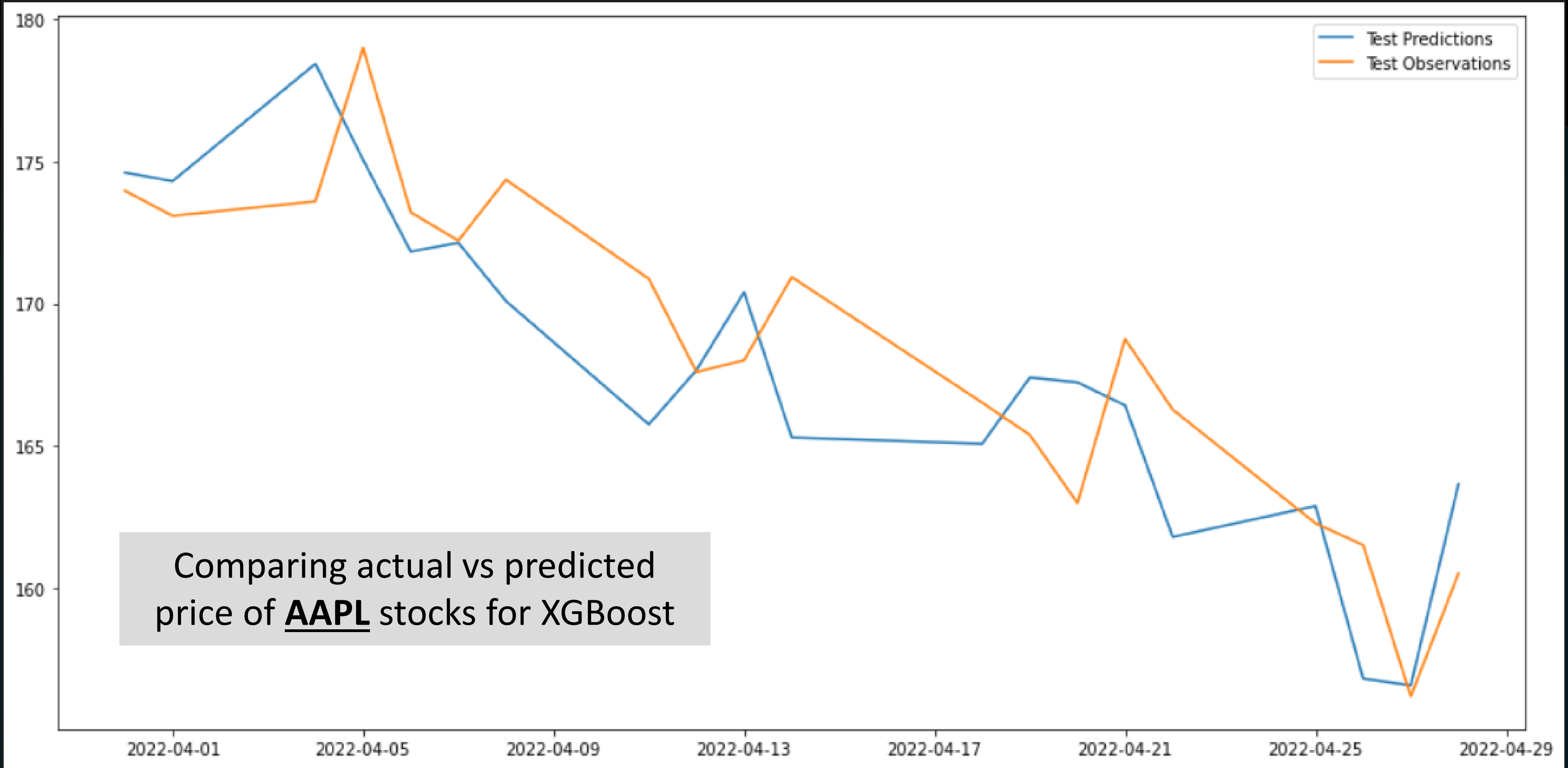
Tuned performance



```
The RMSE for test set is: 0.16358417247864748
The MAPE for test set is: 0.014566397556358605
The R2 Score for test set is: 0.8782150466987036
```

- Best Params was used but found that base model performance was still better than the suggested best params

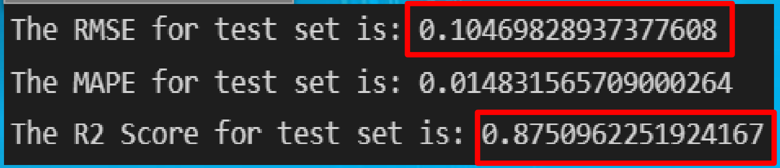
- Base model was used for forward testing



Comparing actual vs predicted price of **AAPL** stocks for XGBoost

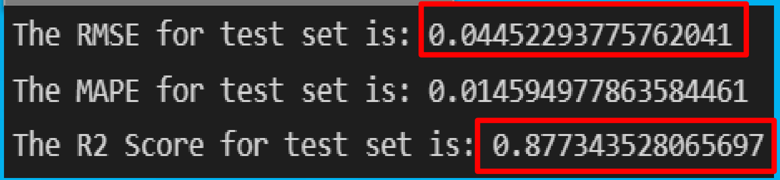
LSTM

Base Model



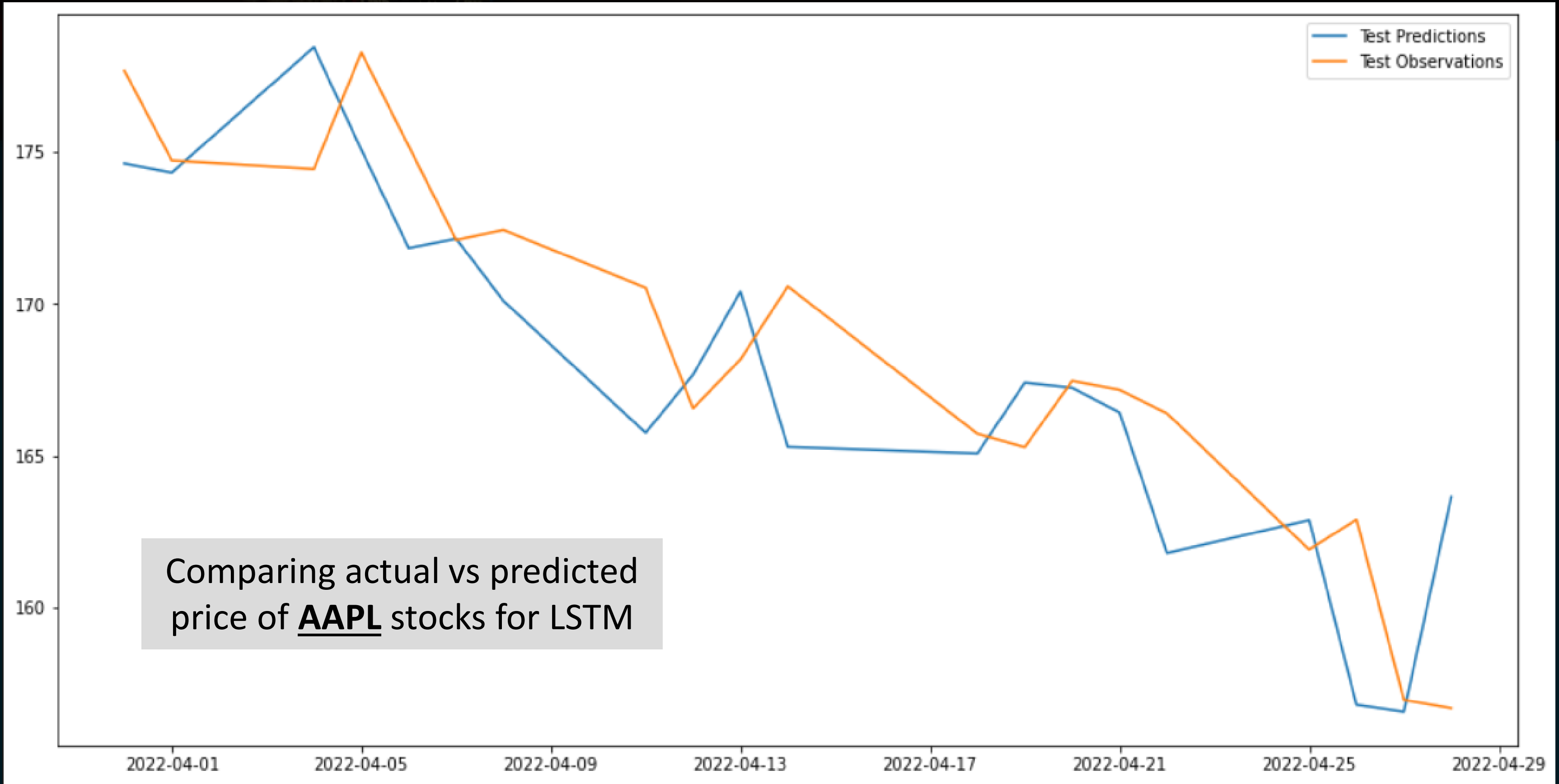
```
The RMSE for test set is: 0.10469828937377608
The MAPE for test set is: 0.014831565709000264
The R2 Score for test set is: 0.8750962251924167
```

Tuned performance



```
The RMSE for test set is: 0.04452293775762041
The MAPE for test set is: 0.014594977863584461
The R2 Score for test set is: 0.877343528065697
```

- Best parameters LSTM got better performance as such the tuned LSTM was selected

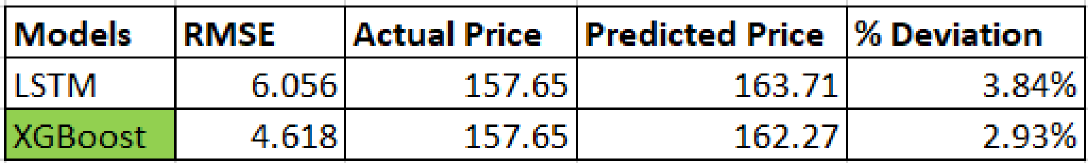


Comparing actual vs predicted price of **AAPL** stocks for LSTM

Forward Testing of models

Predicting price for 29th April 2022

Let's see **AAPL** since we used it for our training……



Models	RMSE	Actual Price	Predicted Price	% Deviation
LSTM	6.056	157.65	163.71	3.84%
XGBoost	4.618	157.65	162.27	2.93%

XGBoost seems to perform better…

Forward Testing of models

Predicting price for 29th April 2022

Let’s see how the model(s) does for other stocks……

AMD

Models	RMSE	Actual Price	Predicted Price	% Deviation
LSTM	4.223	85.52	89.74	4.94%
XGBoost	4.998	85.52	90.52	5.84%

BAC

Models	RMSE	Actual Price	Predicted Price	% Deviation
LSTM	1.161	35.68	36.84	3.25%
XGBoost	0.987	35.68	36.67	2.77%

PFE

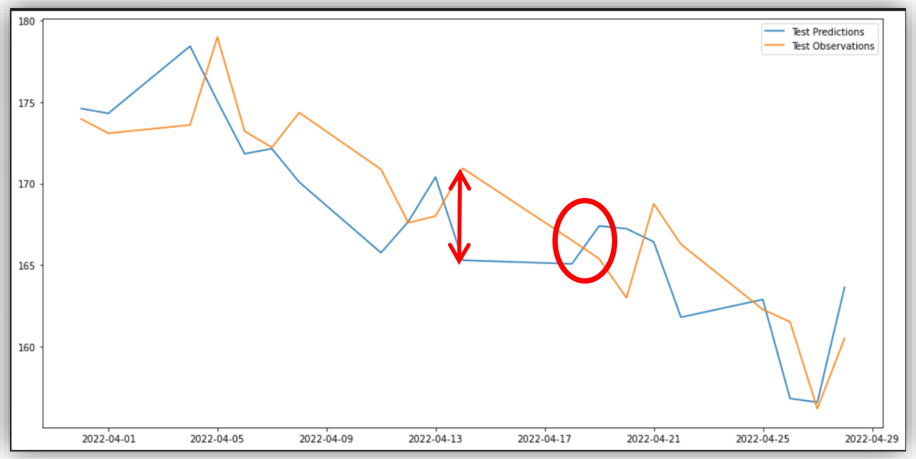
Models	RMSE	Actual Price	Predicted Price	% Deviation
LSTM	1.468	49.07	50.54	2.99%
XGBoost	0.3	49.07	49.37	0.61%

Insights

- Intended to predict 3-day in advance but synthetic data will be used which is not accurate
- Model was unable to take into account news, trader sentiments and black swan events when predicting prices
  - Netflix price crash on 20th April 2022
  - 347.99 (predicted) vs 226.19 (Actual)

Insights

- Not viable for actual trading
  - Wide deviations
  - Opp price direction predicted
- Perhaps to look into including sentiment analysis or used ML to predict trade actions based on strategy



GitHub Link

- https://github.com/edowin25/stockprediction