QFin Momentum Trading Report

Trading team 2

Sem 2 2022

1 Introduction

1.1 Range Trading

Range trading is a trading strategy that identifies a range at which the investor buys and sells at over a short period. When markets are trending sideways instead of upwards or downwards, it is called range. Range trading usually occurs in trendless markets(range). In a range the prices are confined within boundaries. The lower boundary is called support and the upper part of the boundary is called resistance. The goal of range trading is to utilize these indicators/boundaries to maximise profit.

1.2 Exploring Indicators

Out of various indicators, we have primarily focused on researching Bollinger Band and Average True Range. The main exploration is to visualise the price with the indicator and to adjust the parameters to get some understanding on how the indicator work.

1.2.1 Bollinger Bands

Bollinger Bands are two boundaries (top and bottom) plotted at a standard deviation level above and below a simple moving average of the price. Since the distance of the bands is based on standard deviation, they can be used to predict volatility and other measurements. There are three main parts of the Bollinger band, the upper band, lower band, and the middle band (default 20 days moving average).

When trading with the Bollinger band, it is highly advised to stay focus on the trend alongside trading with the support and resistance according to the upper/lower band. Traders can utilize the moving average to know the ongoing trend.

One of the ways to use the Bollinger bands is through the ‘squeezing’ technique, which predicts that there will very likely be a breakout when the bands have been tight for a period.

To test this out, we try to visualise the outbreak with matplotlib.

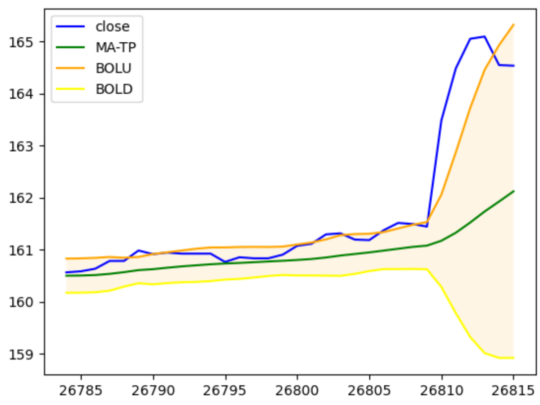


Figure 1 Visualise breakout with Bollinger band

1.2.2 Average True Range (ATR)

Average True Range (ATR) is the average of true ranges over the specified period. ATR measures volatility, considering any gaps in the price movement. Typically, the ATR calculation is based on 14 periods, which can be intraday, daily, weekly, or monthly. There are 3 variables needed for ATR – high low, high close and low close.

$$High\_low = High\ price - Low\ price$$

$$High\_close = High\ price - Close\ price$$

$$Low\_close = Low\ price - Close\ price$$

$$TR = MAX[(High\_low), ABS(High\_close), abs(Low\_close)]$$

$$ATR = \frac{1}{n}\sum_{i=1}^{n} TR_i$$

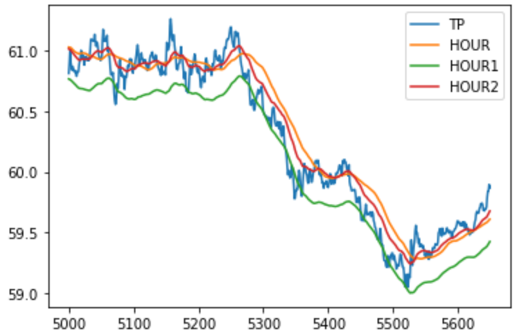


Figure 2 Visualising ATR

1.3 Nature of the data

We have also tried to visualise how different time intervals (i.e., 1 minute, 15 minutes, 60 minutes) affect the volatility of the prices and compare them. We realised for shorter intervals we will get to observe the volatility and outbreak more (since the training period is affected by the time interval) but there will also be more noise compared to longer intervals.

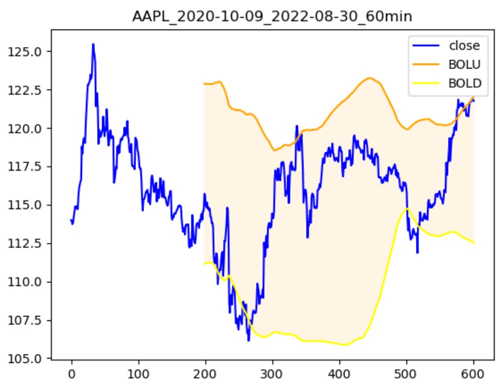


Figure 3 AAPL visualisation (60 minutes interval)

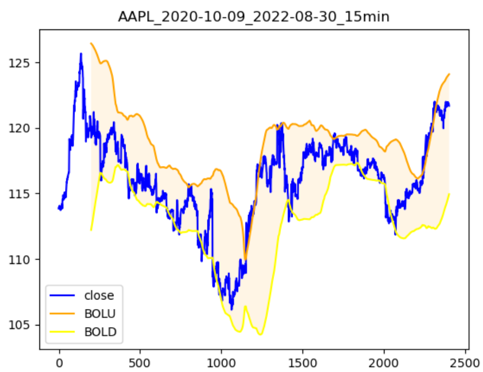


Figure 4 AAPL visualisation (15 minutes interval)

2 Machine Learning & Feature engineering

2.1 Transition to Machine Learning

Although we have tried to trade with the Bollinger bands indicator, we realise there are still a lot of space for improving our algorithm, so we decided to try out other trading methods.

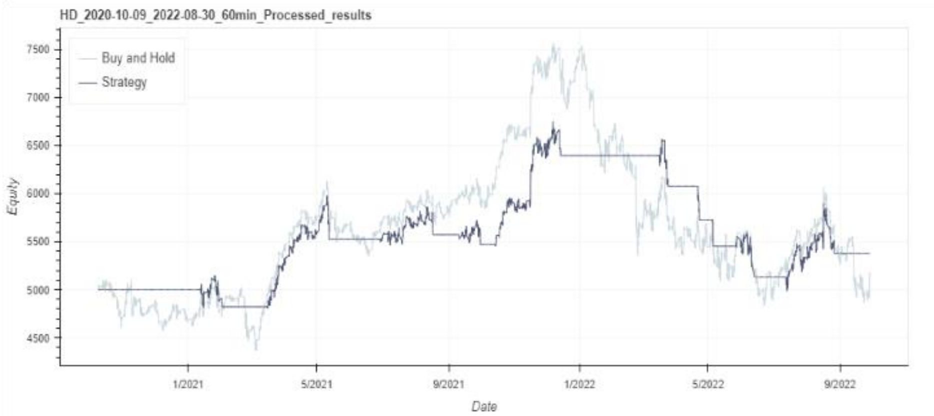


Figure 5 Trading HD stocks using Bollinger bands

Machine learning (ML) is the use and development of computer systems that can learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data. Transition from traditional trading method to machine learning required the understanding of feature engineering and different machine learning models. We have chosen linear regression as our machine learning model due to simplicity and easy to approach compared to other models.

2.2 Linear Regression

Linear regression is a supervised Machine Learning model (already have built in libraries) in which the model finds the best fit linear line between the independent(x) and dependent variable(y). To apply the linear regression model into coding(python), we used the sklearn library and XGBRegressor from xgboost.

$$y = \theta_1 + \theta_2 x$$

To check the accuracy of the newly trained model, we can use mean absolute error (MAE) and root mean square error (RMSE) to measure the error of the prediction versus the real outcome.

Different measurements can be used to enhance the function of the model. For example, we can apply regularization, which reduce the model’s overfitting or improve the underfitting of the model. Elastic Net is one of the most used models that optimise Ridge and Lasso model and performs variable selection and regularization simultaneously.

3 Algorithm

3.1 Using ATR indicators for feature engineering (Data-pre-processing)

```
    factors = [
        '2ATR15Min_wav', '2ATRHour_wav',
        '5Candle', '50Candle']

    # BASIC DATA
    df['15Min_wav'] = df['close'].ewm(alpha = 0.1).mean()
    df['Hour_wav'] = df['close'].ewm(alpha = 0.03).mean()
    df['Day_wav'] = df['close'].ewm(alpha = 0.0015).mean()
    df['5Day_wav'] = df['close'].ewm(alpha = 0.0003).mean()

    df['%15Min_wav'] = (df['close'] - df['15Min_wav'])/df['close']
    df['%Hour_wav'] = (df['close'] - df['Hour_wav'])/df['close']
    df['%Day_wav'] = (df['close'] - df['Day_wav'])/df['close']
    df['%5Day_wav'] = (df['close'] - df['5Day_wav'])/df['close']


    df['Candle'] = (df['close'] - df['open'] > 0)*2 - 1
    df['5Candle'] = df['Candle'].rolling(5).sum()
    df['50Candle'] = df['Candle'].rolling(50).sum()
    df['1000Candle'] = df['Candle'].rolling(1000).sum()


    high_low = df['high'] - df['low']
    high_close = np.abs(df['high'] - df['close'].shift())
    low_close = np.abs(df['low'] - df['close'].shift())
    ranges = pd.concat([high_low, high_close, low_close], axis=1)
    true_range = np.max(ranges, axis=1)
    df['1ATR'] = true_range.rolling(100).sum()/10
    df['2ATR'] = true_range.rolling(10000).sum()/1000


    df['1ATR15Min_wav'] = (df['close'] - df['15Min_wav'])/df['1ATR']
    df['1ATRHour_wav'] = (df['close'] - df['Hour_wav'])/df['1ATR']
    df['1ATRDay_wav'] = (df['close'] - df['Day_wav'])/df['1ATR']
    df['1ATR5Day_wav'] = (df['close'] - df['5Day_wav'])/df['1ATR']

    df['2ATR15Min_wav'] = (df['close'] - df['15Min_wav'])/df['2ATR']
    df['2ATRHour_wav'] = (df['close'] - df['Hour_wav'])/df['2ATR']
    df['2ATRDay_wav'] = (df['close'] - df['Day_wav'])/df['2ATR']
    df['2ATR5Day_wav'] = (df['close'] - df['5Day_wav'])/df['2ATR']
```

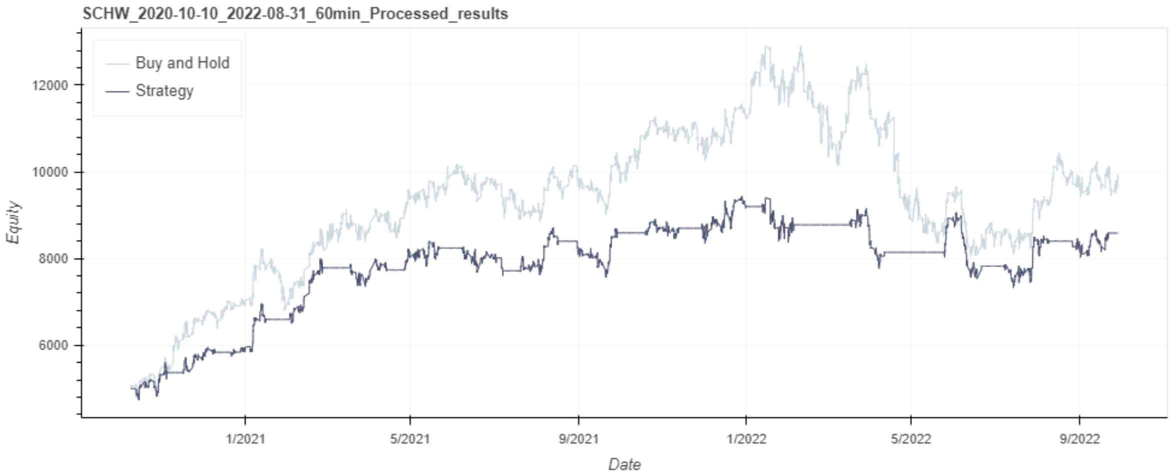
The features/factors that were used for feature engineering are 2ATR15Min_wav, 2ATRHour_wav, 5Candle, 50Candle. A candle(stick) is a type of price chart used to displays the high, low, open, and closing prices of stocks for a specific period. It is calculated by subtracting the open price from the close price, multiple by 2 and subtract one. The number in front of the ATR is used to distinguish the rolling period (i.e. 2ATR equates to rolling 10000)

3.2 Training model with data

```
X = df[factors]
y = df['return']

X_train, X_valid, y_train, y_valid = train_test_split(X, y)

model = XGBRegressor(learning_rate = 0.025, n_estimators=300) #random_state=0, learning_rate = rate, n_estimators=500
model.fit(X_train, y_train,
          eval_set=[(X_valid, y_valid)],
          verbose=False,
          eval_metric='mae')
predictions = model.predict(X_valid)

mean_absolute_error(predictions, y_valid)

model.save_model("model_sklearn.json")
```

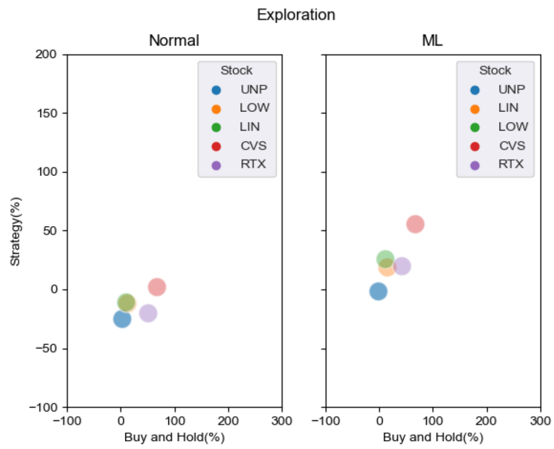
4 Back-testing

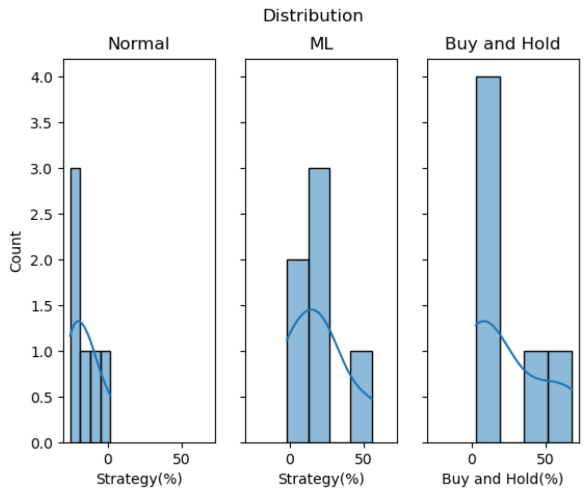
4.1 Back-testing

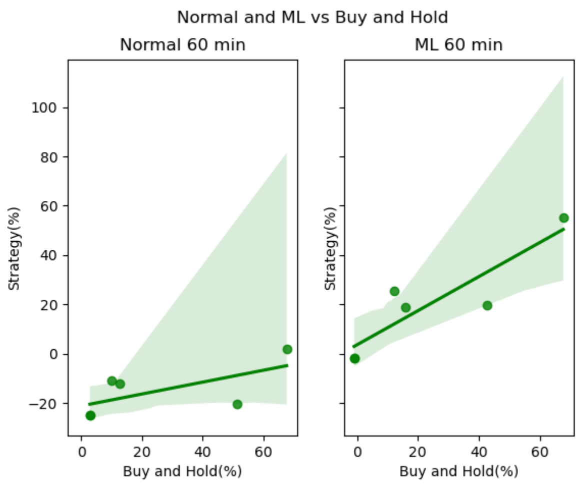


4.2 Analysis results

According to the results shown above, our algorithm is consistently profiting. However, the profits from each stock do vary greatly, from 1.86% to 71.73%. This could mean that our algorithm is still not very stable and if given more time should back-test more and adjust the parameters. We then use 6 stocks to do extra analysis. Neither of the old or the new ML model performs better than Buy and Hold but the machine learning algorithm performs still performs well and have no loses.







4.3 Improvement

To Improve our algorithm, one of the best ways is to add some risk management to our algorithm. This could be limiting the percentage of money that can be used once the buying power drops to a certain level. Another more straightforward way is just to get more training data.

6 Limitations and Challenges

6.1 Challenge – Feature engineering

One of the most challenging aspects when applying machine learning is feature engineering. Extracting the correct data as features are crucial on whether the training will be useful and efficient. Furthermore, feature engineering requires both understanding the training model and a decent understanding of the trading discipline and how they work together, which is certainly challenging given the amount of time that we had.

6.3 Limitation - Time for running the algorithm/Crashes

Due to the enormous amount of data being stored in the 1-minute dataset, crashes are not uncommon. Due to the limitations of our machine/laptop, this limitation does stop us from further testing much larger dataset for our algorithm besides 15 minutes or 60 minutes interval.

6.4 Future exploration

Using heatmaps, we hope to explore do more analysis in the standard deviation in the future.