Data science capstone project - Report

# Investment and Trading Capstone Project

## Build a Stock Price directional movement predictor

March 18th 2021.

## Project Definition

### Project Overview

The project idea is to check if it is possible to predict stock directional movement. In simple words, is stock is going up or down. Stock prices are highly volatile non-stationary, and challenging to predict. The objective is to be having better accuracy than 50%. Even slightly better and consistent accuracy with correct trading strategy can create considerable returns. I have taken directional movement prediction (classification problem) rather than stock price prediction (regression). It is more useful to have buy or sell signal than what will be exact price of stock (which may be more difficult to do)

### Problem Statement

To predict price movement direction (Up/Down) for NSE index.

**Metrics** – Accuracy matric is simple choice for binary classification algo. Since results of such algorithums are further feed to backtesting stratagy that will actually can make money in either direction ( in case stock goes up buy and hold , incase stock goes down short sell and buy), we do not need to go further to understand confusion matrix or class specific accuracy. These will be relavant for problems that will need one class accuract more than other.

### Solution strategy

Predicting stock market price direction has been extremely advance topic and anything more than 50% (random) , consistent results will be a good basic solution. It is important to note that stock prices are inherently noisy and it will take multiple alternate data sources along with much advance algo's and infra to get anything near to workable solution. Please note that NSE stock index is most volatile and hard to predict.

### Choice of financial instruments for directional prediction

Intuitively, the choice will be instrument types that can not be manipulated and are purely subjected to market forces. Hence choice made is the NSE ( National stock exchange India Index). Any penny stock or small-cap will be challenging to predict since features may be driven by lot more than what is seen in the data ( market manipulation). Logically commodities like Gold will be more apt in such cases. FX pair may also be appropriate as it will be more stable and less immune to volatility.

**Choice of classifier - Logistic Classifier**

The logistic regression model is a specific type of a broad class of models known as generalized linear models (GLM).

**GLM has three components**

- **Random** - Probability distribution of the response variable (Y). e.g., binomial distribution for Y in the binary logistic regression.
- **Systematic** - Explanatory variables (X1, X2, ... Xk) as a combination of linear predictors; e.g. $\beta 0 + \beta 1x1 + \beta 2x2$
- **Link Function** - the link between random and systematic components – Logit for Logistic regression.

**Fundamentals explained**

In the Linear regression, Y-axis can have any value in Logistic Y is confined to a probability between 0 and 1. Hence Y axis in logistic regression is transformed from probability to log, so, just like the y-axis in linear regression, it can go from -infinity to + infinity. We do this transformation with a logit function log(p/(1-P). The midpoint of the Y-axis with probability 0.5 maps to 0 on the transformed axis, p = 1 mapping to + infinity, and P=0 mapping to -infinity. Like for linear regression y = mx + c

Features with positive coefficients increase the probability of the modeled outcome as they increase, while features with negative coefficients decrease probability.

**Choice of Base line classifier – Dummy Uniform**

Benchmark model will be Dummy Classifier with strategy uniform. It is important to note that volatility will create different periods of up and down move and going with any other strategy may work in testing set, but cannot be garneted with real time data. Specifically if the model needs to be generic to handle intraday 5 min tick data rather than daily data.

**Evaluation Metrics**

Since this is classification problem, Accuracy , Precision , Recall , CNF , TPR , FPR , AUC (class = UpMove) and AUC ( class =DownMove) will be used to evaluate the model. This will be presented for different penalties and compared.

## Analysis

### Data Exploration

Yahoo finance is used as key source of data. After downloading closed prices, additional datapoints are generated to add core as well as lagged features. Various approaches to gain insights to assets under consideration helps to capture as much dimensions in data as possible. Doing univalent analysis using close price only is possible but multivalent approach is expected to give better results. Although pure ticker data ideally should contain most of the information in perfect world, but fact is unless nontraditional sources are augmented and experimented true success may not be possible. Below are the approaches discussed and some of them used further to add new features to original dataset.

| Overlap Studies | Momentum | Volatility |
|---|---|---|
| Exponential MA<br>MACD | CCI<br>Momentum<br>RSI<br>SRSI | Williams<br>ATR |

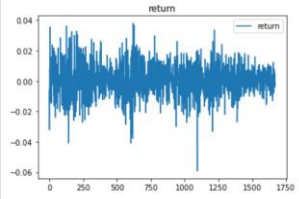
Below is snippet of the final data set.

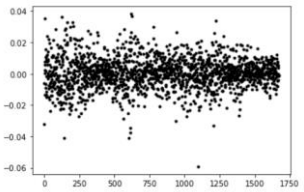
```
#always good to do visual check
data.head()
```

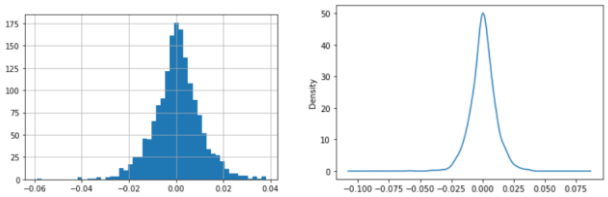
| | Unnamed: 0 | Class | return | ret_1 | ret_2 | ret_3 | ret_4 | ret_5 | return_sign | EMA_12 | EMA_26 | 10 Day ROI | 20 Day ROI | 30 Day ROI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | -0.005824 | -0.008952 | 0.010927 | -0.015776 | 0.011812 | 0.000128 | -1.0 | 5458.284660 | 5502.033202 | 0.034986 | -0.043966 | -0.055048 |
| **1** | 1 | 0 | -0.032120 | -0.005824 | -0.008952 | 0.010927 | -0.015776 | 0.011812 | -1.0 | 5428.194742 | 5484.304831 | 0.007061 | -0.060953 | -0.102426 |
| **2** | 2 | 1 | 0.007762 | -0.032120 | -0.005824 | -0.008952 | 0.010927 | -0.015776 | 1.0 | 5409.018598 | 5470.915570 | -0.001215 | -0.037844 | -0.077948 |
| **3** | 3 | 1 | 0.005600 | 0.007762 | -0.032120 | -0.005824 | -0.008952 | 0.010927 | 1.0 | 5397.361891 | 5460.718120 | -0.022498 | -0.031357 | -0.056821 |
| **4** | 4 | 1 | 0.035447 | 0.005600 | 0.007762 | -0.032120 | -0.005824 | -0.008952 | 1.0 | 5416.583108 | 5465.279727 | 0.007535 | 0.019401 | -0.023423 |

### Data Visualization

We are interested in nature of returns and hence it will be important to look at them closely.

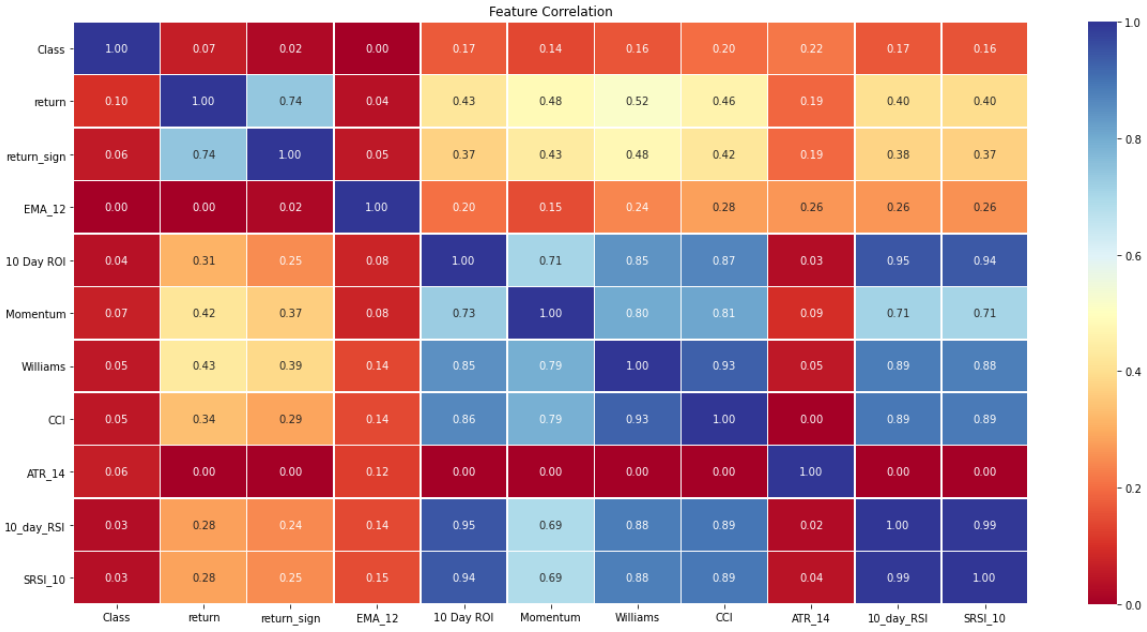





Returns clearly are not normally distributed. To understand the co-relation lets start with basic corelation plot and then we will explore log of features to transform the non normal data set to near normal and understand the co-relation better. This will help us to select features that can be further used in the model.

Before log transform

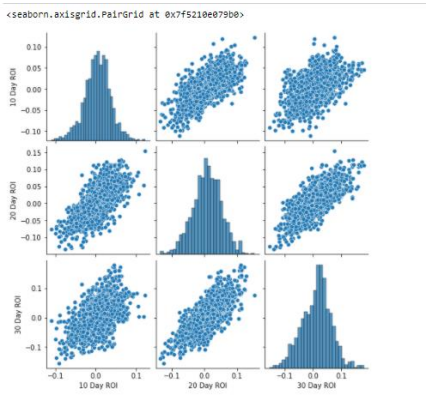



All features seem to have less co-relation with class output. 10 Day ROI , Momentum , Williams and CCI are hightly corelated. Same with SRSI_10 and RSI_10. Based on this we can select one from each set that are co-related. Before we do that , lets understand the corelation for lagged features and what do these indicator mean and which one we can drop. Although returns and return lags seem to have lot of information, since taking return will snikein next day data, we will need to drop that.

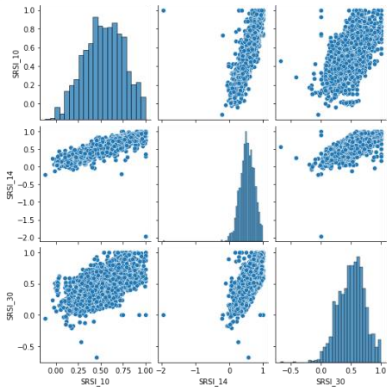
Pair plots are good way to understand the relation between lagged features. If pair plots show linear relation between two laggs, we can select one of them. If the relation is more noise then we can conclude we have some new information or dimention that can be used. Since time series data is auto regressive, taking lags for 10 , 15, 30  day will give some more insights to the data.

Below are the key plots

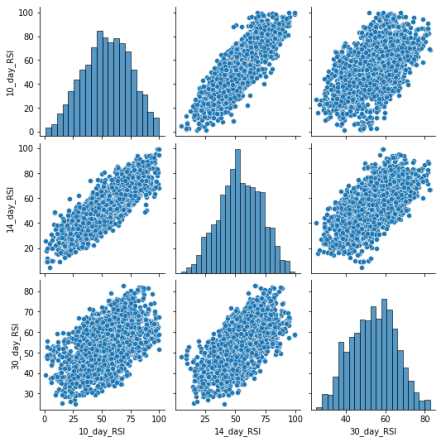
ROI




SRSI




RSI



Based on all the above results and domain understanding we have about domain it will be safe to take below final features.

'Momentum','10_day_RSI','30_day_RSI','MACD_12_26','SRSI_10','SRSI_14','CCI'

Typically momentum and overlap studies indicators are important and laggs capture critical data.

### Algorithms and Techniques

Ref Solution strategy

### Benchmark

Benchmark model will be Dummy Classifier with strategy uniform. It is important to note that volatility will create different periods of up and down move and going with any other strategy may work in testing set, but cannot be garneted with real time data. Specifically if the model needs to be generic to handle intraday 5 min tick data rather than daily data.

**Base model accuracy score :** 0.5159362549800797

Base model accuracy score with uniform stratagy with multiple runs does not give any thing better than 50%. Ides is to understand if ML model post training can provide something more consistant and better than 50%.

## Methodology

### Data Preprocessing

Fortunately data is clean and this can be validated by

```
# Check for missing data
data.isnull().sum()
```

```
Class          0
return         0
ret_1          0
ret_2          0
ret_3          0
ret_4          0
ret_5          0
return_sign    0
EMA_12         0
EMA_26         0
10 Day ROI     0
20 Day ROI     0
30 Day ROI     0
Momentum       0
MACD_12_26     0
Williams       0
ATR_14         0
CCI            0
dtype: int64
```

Further to this , prepossing steps needed scaling. I have used MinMaxSacler for the same.

```
#Scale data.
scale = MinMaxScaler()
columns = data.columns
data = scale.fit_transform(data)
```

Apart form these two steps , we did not need any other preprossing. ( Note : Adding new features is not treaded as preprocessing and was discribed under feature enginering section )

## Implementation

**Selection of algorithm** – Please ref to - **Choice of classifier - Logistic Classifier**

Since dats is slightly imbalanced the weights are adjusted accordingly.

```
weights = {0:1.1, 1:1.0}
model = LogisticRegression(solver='lbfgs', class_weight=weights , C = 1e5,max_iter=1000)
```

Logistic regression is basic model and many advance models and additional and advance features can be used. This topic is consistantly under research. It will be interesting to add advance features and use alternative data sources and experiment with DL models , specially RNN's and auto encoders. Due to serious time constrains these areas are not explored but indicated to explain further possibilities.

**Selection of metrics** – Accuracy matric is simple choice for binary classification algo. Since results of such algorithums are further feed to backtesting stratagy that will actually can make money in either direction ( in case stock goes up buy and hold , incase stock goes down short sell and buy), we do not need to go further to understand confusion matrix or class specific accuracy. These will be relavant for problems that will need one class accuract more than other.

## Results

### Model Evaluation and Validation / Justification.

As indicated, since trading strategies can benefit from both class 1 and class 0 accuracy, simple measure of accuracy is enough. This essentially means in either direction, we will be able to make money as explained earlier.

Final model accuracy is much better than random model. Please note that given that we have very less signal and more noise, expectation is , anything better than 50% consistently will provide good basis to create trading strategy that can create profits. With approaches like Kelly optimal bets ( only part of the cash is invested in the ration to the confidence / probabilities of each class) we can safely make profits and avoid deep negative losses.

```
Final model accuracy 0.5517928286852589
```

## Conclusion

## Reflection

We started with finding interesting data science problem to solve. Given that I am from finance background and recently completed my CQF certification 😊, I wanted to understand in principle how ML can be applied to stock prediction.

Mathematical vs ML models
Quantitative finance traditionally has used mathematical modeling approach. Although this is good approach for problems when we have smaller datasets and ML is difficult to apply ( credit defaults), some problems like stock price prediction where large tick data is available, it makes sense to use ML models. I like ML models as they are much easier model than mathematical models.

Which problem?
Although it will be ideal to predict actual price , it will be almost impossible to predict exact price. Some of the very basic momentum strategies may give an impression that they are predicting right price but in actuality they are just approximation of recent past. In stock trading models , although simple momentum can be useful to some extend it will be safer to device trading strategy that is based on directional movement. We have little advantage of knowing exact stock price , what is needed is directional movement. This is because we can make money is either direction.

The data and features and feature engineering
Started with freely available data and basic attributes of features. Although univariant model can be made just by using close price, it is possible to engender more interesting features that are based on domain. I used my domain understanding on first selecting key features / technical indicators – some from each group. Calculated values for the same using close price. Please note , data cleanup using various methods would have been required if data was missing but fortunately yahoo provides a clean data. Test for cleanup and approaches are documented at high level.

Data exploration
This step is progressive and is done at almost every step to ensure visually data makes sense. Post basic EDA, advanced EDA that shows co-relation between dependent and independent plus co-linearity within features and most importantly between lagged features was done to understand which features to select from possible feature universe.

It is important to note that if simple co-relation plot does not provide lot of insights we can try log co-relation. For non-linear data it may be interesting to take log rather than simple co-relation.

When we are ready with final features just before we feed it to our model, we must not forget to scale the data.

We then use baseline model – dummy in this case and calculate the accuracy. Then used the final model and while doing to finetune weights (challenge 😊) since our data set is slightly imbalance. Our metric is fairly simple as we are only interested in directional movement – either side is ok.

## Improvement

Many improvements can be done in above model.

Additional alternative data – Typically twitter feed, micro economic data , news etc can be used as additional data source. This is give more features or dimensions. This will need separate models to handle additional data sources to convert them in apt features set.

Feature engineering – We can use SOM / tree based / KNN methods on our features to understand which features are close to each other and then select one of them instead of just using co-relation plots. More interesting will be the distance matric. For time series it will be interesting to use DWT as base input distance matric to our SOM model.

More advanced way for feature engineering will be using auto-encoders. Due to non-linearity in data , neural networks will be ideal for such problem, further to this due to time series nature , LSTM will be apt type of network and finally to reduce the feature set from human understandable to machine friendly features using autoencoders will be the way to move forward on model selection. More advanced modeling techniques using GANs as additional final layer are in latest white papers and we can try them as well. Model that we have used is extremely basic model in comparison to what I have proposed but still fundamental to understanding of ML models.