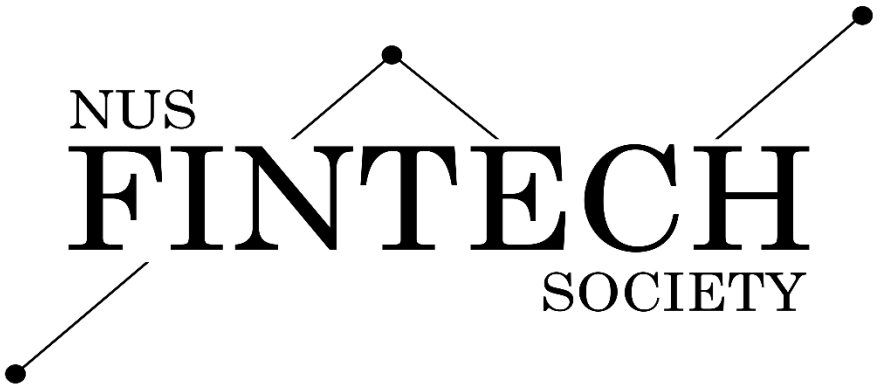

Logistic Regression RSI Trading Strategy

Authors

Chiran Bopitiya | Chloe Tan | Lim Zi Yun

nusfintech@gmail.com

SUMMARY

In this project, we attempt to create the best model for trading in the Forex market, in the period of 7/11/2018 to 20/11/2018, using the Oanda brokerage. The trading model that we selected was a Logistic Regression model with an L1 penalty, supported by the Relative Strength Index technical indicator to help us trade the EUR/USD currency pair in the Forex market. The training of the model was done by retrieving the past 50 days of EUR/USD and using a 10 day lag period, as well as selecting 3 other currency pairs that we deemed to affect the EUR/USD movements the most based on our research and numerous trials, namely the GBP/JPY, USD/JPY and USD/CHF. The above selection of features were chosen as it gave us higher overall returns and accuracy of the training model after numerous rounds of backtesting. Reinforcement learning was also adopted to retrain our model in the case of wrong predictions, allowing our model to increase its returns. This report covers how we decided on our trading model and its features, and provides some future improvements for our model.

TABLE OF CONTENTS

MAIN REPORT

1. INTRODUCTION

In this project, we are interested in creating a model for “live” trading for the period of 7/11/2018 to 20/11/2018, which will give us the highest return. The model that we are interested in creating will generate the direction of trade, and according to the buying and selling conditions set, the model will be making trading decisions for us.

2. SELECTION OF DATA AND ASSETS

We are using Oanda Brokerage and since the brokerage focuses mainly on the Forex market, we decided to trade in the Forex market. We also chose EUR/USD as our trading currency as it is one of the most popular currencies being traded in the Forex market. The dataset is obtained from QuantConnect’s library of data for the Forex market. We used the 15 day period from 15/10/2018 to 30/10/2018 to backtest and simulate the 2 week period that we will be doing “live” trading. In this project, we are only looking at the “close” price. We also made use of the Relative Strength Index technical indicator to help us trade.

3. SELECTION OF TRADING MODEL

We decided to go with a logistic regression model for our machine learning strategy as compared to other strategies (such as LSTM, NN) due to our worry of overfitting from these complex neural networks. We felt that neural networks were not very suitable for trading algorithms in this scenario due to the following reasons:

Overfitting is one of the most prominent problems in financial time series forecasting, due to reasons such as the non-linearity of the underlying relationships between time and the exchange rate, and a very strong dependency of the hyperparameters on our choice of model. For the case of neural networks, we will need to fine-tune numerous hyperparameters such as number of hidden layers, hidden neurons, learning rate, the number of iterations and activation function, on top of deciding on how many days of past data the model should take in for training. Therefore, there will be a high tendency for many people to alter their hyperparameters based on the performance of their

models using the same sample data, which will inevitably lead to overfitting of noise and biased results.

Hence, we decided to adopt a logistic regression model instead because logistic regression helps measure the relationship between the dependent variable (our label we want to predict) and the one or more independent variables (our features), by estimating probabilities using it's underlying logistic function. The logistic regression model is efficient and does not require too many computational resources, does not require input features to be scaled and it is easy to regularize.

4. BUYING/SELLING DECISION MAKING PROCESS

Our model first makes an UP or DOWN prediction, that signals to us the direction the model believes the exchange rate will move in. If the model signals UP, we will check that the current RSI value is less than 30, before buying the currency using 90% of our holdings and add EUR/USD to our long list. If the model signals DOWN, we check if the RSI value is more than 70 and then take a short position for the currency pair, similarly using 90% of our holdings, and add the pair to our short list.

5. RISK MANAGEMENT

We set a Stop Loss - Take Profit (SL-TP) range so that we know when to minimize our losses or gather our profits. This is to minimize losses in the event that our predictions are wrong (stop loss), and to prevent further exposing our entire account equity to further market fluctuations, which may turn the tides of our profits if there is a sudden change in direction of the currency (take profit). This stop-loss take-profit margin is set at 1% in each direction.

6. MODEL EXPERIMENTATION AND RESULTS

There are a few parameters in our model which could be varied. The parameters were tested one by one, with all else being constant.

Firstly, we included various currency pairs to test for correlations, which might affect the prediction of the direction of our trading currency. The currency pairs include GBPJPY, USDJPY, USDCHF, USDGBP.

The currencies GBPJPY, USDJPY and USDCHF gave us the highest returns when testing and hence are the only currencies that were included in our final model.

Secondly, in our logistic regression model, we varied the hyperparameter “penalty”, testing using L1 and L2 regularisation. “L1” penalty gave us higher returns than “L2” and hence in our logistic regression model, we used “L1” for the hyperparameter penalty.

Thirdly, for training the model, we varied the number of data points by using different number of days of historic data. We tested for 500, 300, 100, 75, 50, 30 and 25 days. 50 days of historic data gave the best returns. Through varying the different number of days for historic data, we can infer that a lower number of days used to train the model can help produce better results as a higher number of days may introduce irrelevant and outdated information into our model. However, too low a number causes the model to perform poorly as there could be too little information for the model. In this case, 50 days gave the highest accuracy and returns.

Lastly, we tested different number of days for the number of lags to use for training: 10, 15, 20 and 30 days. A 10 day lag period gave us the highest returns.

We also attempted **reinforcement learning to retrain the model should our model predict wrongly.** Upon a wrong prediction, we will retrieve the latest data using the self.history function. This will include the more recent days of data and exclude the older data, therefore making the data more relevant in predicting. The model is then re-trained using the same features and it will replace the old model. It is done as soon as there is a wrong prediction, and we will immediately use the new model to make a new prediction so that the immediate next day will utilize the new model to predict.
We tested 2 different models, one with reinforcement learning and without on an arbitrary time period of 15 days to simulate the live testing period duration of 14 days from 7th Nov to 21st Nov

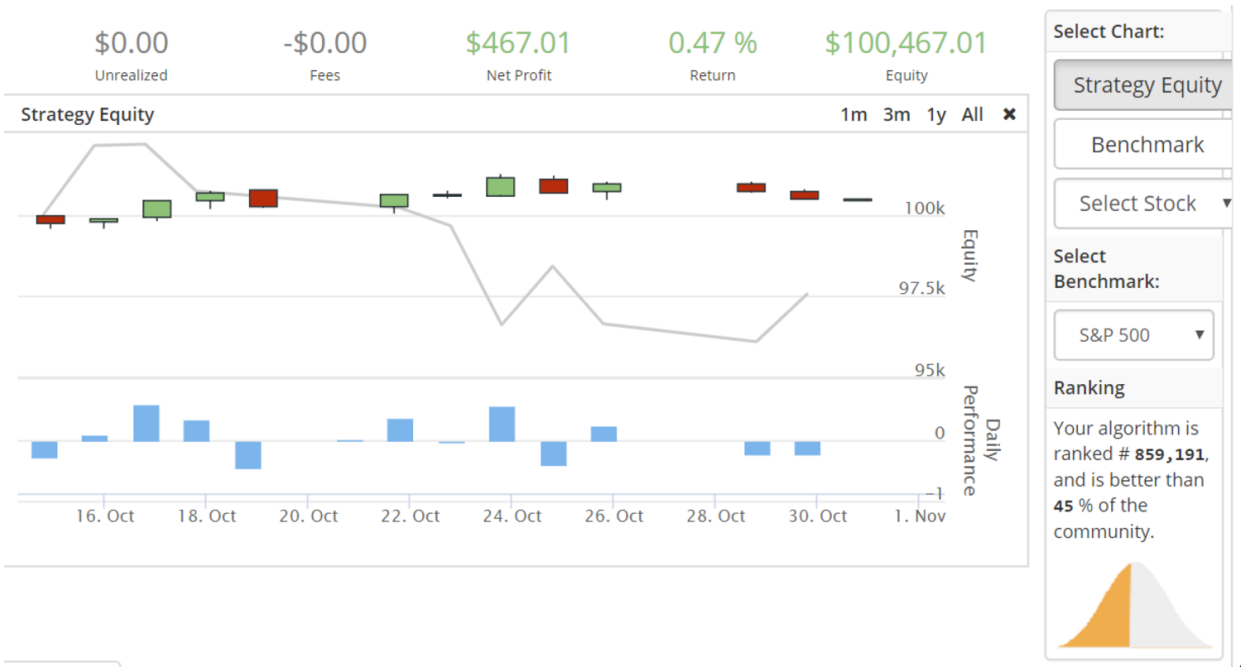

Fig 1: Model performance for 15 day period 15 Oct 2018 to 30 Oct 2018 without reinforcement learning

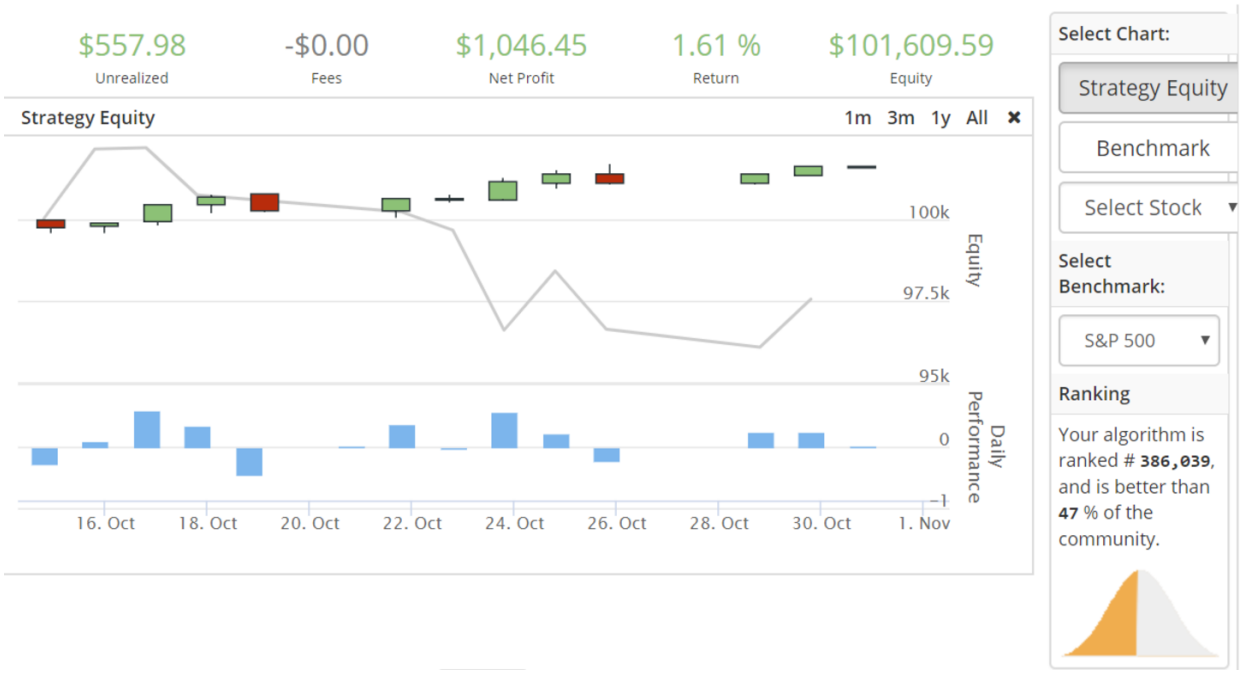

Fig 2: Model performance for 15 day period 15 Oct 2018 to 30 Oct 2018 with reinforcement learning

7. PERFORMANCE MEASUREMENT

Overall, we were less concerned about the accuracy of our trained model as compared to the actual backtest. Knowing that financial time series analysis usually comprises numerous non-linear relationships, a high accuracy (80% and above) does not necessarily mean that the model will do well on actual data. A low accuracy (50% and below) does however, imply that we might need to reconsider some of the features of our model. We used the self.accuracy function on our logistic regression model and after fine-tuning our features and hyperparameters, we were able to **consistently achieve around 65% to 75% accuracy** on numerous rounds of backtesting, and were able to **consistently achieve a positive net profit at the end of the backtest**, indicating that our model is not overfitted and can perform well on actual data.

8. FINAL MODEL

After the experimenting, our final model is a logistic regression model with “L1” penalty (L2 regularisation) with three currency pairs GBPJPY, USDJPY, and USDCHF. We used 50 days of historic data and 10 days lag period for training our model.

9. FURTHER TESTING

3 major decisions made by your model in back testing

3 key decisions that our model made

Fig 3: Model Performance for period 7/11/2018 to 21/11/2018 (Backtest Results)

Evaluation of model decisions:

From our Trades Summary in Fig 3, we bought the currency as our model made a decision to take a long position on 7/11/2018 since the prediction was UP. But from the stockplot we see that the price went down instead. We reached our stop loss of 1% on the 9th and exited the long position. For the subsequent days, the RSI condition was not met, hence we did not enter any trade positions. On the 19th Nov, the model predicted down and RSI condition was met, hence we entered a short position.

10. EVALUATION AND FUTURE IMPROVEMENTS

a) STRENGTHS & WEAKNESSES

Strengths:

The model was not solely dependent on the Logistic Regression output to make a decision. As previously seen in the backtest in Fig 3, there was a period where no trades were made as we considered another technical indicator as well (RSI). Hence, one strength of our overall model would be the complementation of machine learning algorithms with other technical indicators to make informed decisions.

Weaknesses:

Lack of ensemble methods could have caused some overfitting of the model

Reinforcement learning was not able to apply more weight on the recent day's data. Model was very dependent on past and historic data. If there were a time period now that was never seen before in the past, our model would have been unable to handle the input.

b) FUTURE IMPROVEMENTS/CONSIDERATIONS

A **trailing stop loss** where the stop-loss and take-profit cut off value could scale in the direction which we predicted could be implemented. For example, when our model accurately predicted UP, and after taking a long position, we find that the rate is indeed increasing, then we might want to increase both our stop-loss and take-profit values. In this way, we can take advantage of this trend and "lock in" profits as the price moves in our favor.

We could perhaps also take a look at other prices provided, such as the open, high, low prices, and use them in our model.

REFERENCES

The Logistic Regression Algorithm. (2018, April 23). Retrieved from https://machinelearning-blog.com/2018/04/23/logistic-regression-101/