

Final Presentation

Presenter:- Divya Gupta (dg622r)

<https://github.com/VISHDIVGAU/ForexData>



Intelligent Platform for currency trading

- We had built a software platform for currency trading where we can predict returns for next six minutes.
- Using sum of actual returns and future returns in an hour, We can made decisions of investment or no investment in that hour.
- Decision is based on following rules:-
 - If predicted return and actual return are divergent meaning for example if we are having loss in actual return and having profit in predicted return or vice versa. We will not invest.
 - If predicted return and actual return are convergent, then we will calculate the percentage of difference between actual and predicted return. For currency we are buying, if the profit or loss from actual returns and predicted returns are negative, meaning we are incurring loss and percentage of difference is greater than 0.25 in first hour, we will not invest.
 - Similarly for currency we are selling, if profit and loss from actual return and predicted returns are positive and greater than 0.25 in first hour, we will not invest
 - We will calculate the percentage difference each hour henceforth from 0.25 to 0.15 to 0.10 to 0.5. We will not tolerate loss greater than 0.5 after fourth hour.
 - If none of the above conditions applies, we will invest 100 value in that hour.

Improvements

- I am now using MongoDB for storing data.
- I have collected approximately 275 data points to train my regression model in database "ForexData_project".
- I have normalized FD, Volatility and returns for better classification.
- I have included day, hour, minute as features in my model to capture the changes with day, hour and minutes in my model.
- I have three models for each currency pair classifying FD and volatility differently using three different sorting method.
- I am saving model with least R^2 value for each currency pair.
- Similarly, I am saving best classification method for each currency pair, so that we can classify real time data of that currency pair accordingly.

Data Collection

- Run "trailingstops_project_data_main.py" to collect data.
- It will create _agg tables for each currency pair and store aggregate data over six minutes in these tables.
- Following fields are being stored in these tables.
 - Inserttime
 - Avgfxrate
 - min_price
 - max_price
 - Volatility
 - fractal_dimension
 - return

Data preprocessing steps

- We require some data preprocessing steps for better predictions of return values from our regression model.
- Some of the steps done are:-
 - Remove entries with volatility 0 while sorting, classify and training models. This is the data after Friday 5pm, after markets got closed.
 - Multiply return value by 1000000
 - Multiply volatility by 10000
 - Divide fractal_dimension by 1000000

Training Model

- Use `trailingstopsproject_training_model_main.py` to train your regression model.
- It will use the data stored in `_agg` tables in `FrorexData_project` database.
- It will first call `sort_one`, `sort_two` and `sort_three` methods.
- These methods implements different sorting methods of sorting FD and volatility column.
- We will store min and max range for six partitions in `sort_one` method and nine partitions in `sort_two` and `sort_three` methods in `_sort_one`, `_sort_two` and `_sort_three` collections for each currency pairs.
- We will call `classify()` method. In this method we will create list of data frames for each currency pairs classified using above three sorting methods.
- In train model, we will create three models for each currency pairs.
- We will select the model with minimim R2 value and save its index in `best_classification` collection.

Prediction on real Time Data

- We will use `trailingstopsproject_realtimedata_main.py` to get predictions on real time data.
- We will use data in `_sort_one`, `_sort_two`, `_sort_three` and `best_classification` collections in `ForexData_project` database.
- In `aggregate_raw_data_tables()` method along with calculating FD, Volatility and return in this function, we will get predicted returns for next six minutes.
- For this we need to classify the current aggregated data using best classification method and pass the data frame to `predict_real_time_return()` method.
- This method will load the saved best model for that currency pair and return the predicted returns.
- We will save all this data in `_agg` collection for each currency pair in `ForexDataProjectRealTime` database.

Continued...

- Following data is saved in _agg collections
 - Inserttime
 - Avgfxrate
 - min_price
 - max_price
 - Volatility
 - fractal_dimension
 - Return
 - future_predicted_return
 - predicted_return
 - Error

Continued...

- At the end of first hour, we will call `layer_one` function.
- In this function, we are aggregating return and `future_predicted_return` and calculating actual return in current one hour window and future return in next one hour window.
- We are calculating `profit_loss` for both current and future returns and difference between future and current returns.
- We will make the decision of investing 100 value more based on the rules discuss in the second slide.
- Similarly, at the end of second hour, we will call `layer_two` method, at the end of third hour we will call `layer_three` method and at the end of fourth hour and so we will call `layer_four` method.
- We are saving `profit_loss` data in `_buy` or `_sell` collections. The last entry in these collections will tell us the total investment we made and total return we got and profit or loss we made in 10 hour window.

Continued..

- We are saving following data in `_buy` or `_sell` collections if that currency pair lies in buying list or or selling list respectively.
 - Inserttime
 - initial_investment
 - agg_return
 - agg_predicted_return
 - actual_profit_loss
 - predicted_profit_loss
 - prediction_error

Thank you

