**LAZY TRADING PROCEDURE**

The timing for this procedure is made for a 5 minutes period.

# VPS Installation

* Copy Lazy trading project folder
* Install google chrome
* Install notepad++
* Install github desktop
* Install R and R studio (<https://www.r-project.org> - <https://rstudio.com> )
* Install 64 bit Java (<https://www.java.com/en/download/> ) (need for package mailR)
* Install h2o (<http://h2o-release.s3.amazonaws.com/h2o/rel-zermelo/3/index.html> ) (restart R studio after install)
* Install metatrader4
* Copy the install folder to create the 4 terminal and rename the folder to Terminal # ((! the original install can’t be portable)
* Create shortcut of the terminals and paste it in the desktop
* Edit the shortcut target properties adding: \portable
* Log your accounts on each terminal (T1 : live ; T2 : Dev , T3 : PreProd ; T4 : No DSS)

Note : use admiral market demo to be flexible with the funds available

* Edit Generate\_ini.R and run it
* Disable windows update
* Edit Rebootlog.cmd / MetaTraderAutoLaunch.cmd , copy and paste in this following path :  
  “C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup”
* Test MetaTraderAutoLaunch.cmd
* Copy all include files in all terminal
* Add terminal.csv inside MQL4/files folder in each terminal with the terminal number written (let mql4 able to tell in which terminal he is located)
* Tool 🡪 Option 🡪 Email set your email information (in all terminals in order to let watchdog send emails)
* Tool 🡪 Option 🡪 Expert allow DLL

- Run expert Watchdog.ex4, set send email = true to each terminal and keep it running on EURUSD 1M (This tool will automatically **monitor** the status of your MetaTrader’s **connection** to the trade server. If a disconnection occurs, the tool will attempt re-login to the account set up)

* Run datawriter\_v4.02.ex4 with 300 candles and the corresponding indicator “true” used to detect the market type and let it run
* Run datawriter\_v6.01.ex4 with 2200 candles used for selflearning
* Run robot in expert advisor until today to generate OrderResultT2
* Edit all path in All\_path.R

# Macroeconomic event

Prerequisite:

If OrderResultT2.csv is missing, run the MQL4 robot in the expert advisor (until today) in order to generate OrderResultT2.csv

Copy that file into the terminal 3 and rename it with T3

Set your gmail to “less secure app access”

<https://myaccount.google.com/lesssecureapps?pli=1&rapt=AEjHL4NMht6chyrVfyj8a4RbQOeo9mMRNa5YCOLmK9GMwrBthWPD00q5OgXZoCaSkdwu1F5uCOSzeJrEbBfxsKEPgzfoYP8vxQ>

Run the script economicCalendarTrigger.R after the economic calendar downloaded

It generates 01\_MacroeconomicEvent.csv sending the signal to trade or not

Edit path in economicCalendarTrigger.bat

Run economicCalendarTrigger.bat in the task scheduler

# TradeControl

*2 possibilities: using Profit factor or reinforcement learning to allow trade.*

*The following explanation are for reinforcement learning, however Profit factor ( TradeTrigger.R ) is similar*

Run Adapt\_RL\_control\_script.R to generate MangicNumber.rds containing the best control parameter.

Script very slow, frequency of run to be confirmed

Run TradeTriggerRL.r

If need the optimized control parameter alpha, beta , Epsilon are in the following file :

MagicNumber.rds

Tradecontrol will edit: SystemControl1201209.csv (file generated be the robot) , sending the signal to the terminal 3 to trade or not

Edit TradeTriggerRL.bat and set it to the windows task scheduler.

Note :

|  |  |
| --- | --- |
| name | function |
| Adapt\_RL\_control | run write\_control\_parameter() for each system (MN) |
| generate\_RL\_policy | generate the RL policy |
| log\_RL\_progress | retrieve RL model Q values progress |
| record\_policy | save to systemcontrol |
| TradeTriggerRL | allowing trade |
| write\_control\_parameters | create rds file with the best alpha gamma epsilon parameter |

# Selflearning

1. Collect data from Metatrader4.

Run expert DataWriter\_V6.01.mq4 (and keep it always running).

It generates the following files:

AI\_RSIADXAUDCAD5.csv (all pairs)

TickSize\_AI\_RSIADX.csv

1. Generate rds files

Run 01\_CollectData.R each time we want to update the model

It generates the following files:

AI\_RSIADXAUDCAD5.rds (all pairs)

These files will be used to make the model

Edit and set 01\_CollectData.bat scheduler

1. Creation of the model

Run 02\_ForceModelUpdate.R only once to build the model for the first time

It generates the following models:

DL\_Regression-AUDUSD-5 (all pairs)

1. Update the model

Run 03\_TestBuildTestModel.R each time we want update the model, let’s say run it once every 36 bars = 3 hours

It updates DL\_Regression-AUDUSD-5 (all pairs)

It generates the following files:

StrTestFull--EURUSD-5.rds (all pairs)

StrTestFull-EURUSD-5.csv (all pairs)

These 2 files contain data result made by h2o.

It also copies the following file into the R project :

TickSize\_AI\_RSIADX.csv

Edit and set 03\_TestBuildTestModel.bat scheduler

1. Score data

Run 04\_ScoreData.R

It generates the following files inside the master and slave terminal (T1 & T3):

AI\_M5\_ChangeCADCHF.csv (All pairs)

Edit and set 04\_ScoreData.bat scheduler

# Market Type

1. Make model

Run datawriter\_v6.01.ex4 with 12000 candles and the corresponding indicator “true” used to detect the market type. Copy the csv files (AI\_CP5-12000.csv …) generated into the R\_marketType/\_Data Folder.

Select data to build the model running 01\_data\_selection\_5M\_32.R

It generates: macd\_ML5M.rds used to build the model on the next script.

Run again that script only if we want build again a new model from scratch and with a new indicator

1. Build model

Run 02\_BuildModel.R each time we want update the model with new data obtain by a high confidence from a score data.

Generate DL\_Classification\_5M (h2o model)

Can run this file once a day.

Use shiny app inside the folder InspecData to generate macd\_checked.rds (rename file later) and later used in 02\_BuildModel.R binded to macd\_ML5M.rds

1. Detect the market type

Run datawriter\_v4.02.ex4 with 300 candles and the corresponding indicator “true” used to detect the market type and let it run

Run 03\_Score\_Data.R

It detect the market type and generate the following files :

AI\_MarketType\_AUDCAD5.csv (all pair)

Macd\_ai\_classified\_5M32

1. Reinforcement learning on market type

Use reinforcement learning on the detected market type to take the decision if we trust or not the MT

Prerequisite:

The file MarketTypeLog1201214.csv is needed, for the 1st run of MT\_TradeTriggerRL.R, the file doesn’t exist as long the trading system didn’t made any order.

Generate that file using the expert advisor, and copy it into the MQL4/files folder (make sure the EA have taken some orders in order to generate the file)

Run MT\_TradeTriggerRL.R

Generate SystemControlMT1201414.csv in slave terminal (T3)

Edit and set MT\_TradeTriggerRL.bat scheduler

1. Teach the model using the shinny app

# MQL4 Robot

Prerequisite:

AI\_MarketType\_AUDCAD5.csv generate from market type

It generates the following files:

OrdersResultsT2.csv (file written on the first order and updated on each new order)

SystemControl1201209.csv (value generated doesn’t allow to trade)

After each order : MarketTypeLog1201214.csv written (If R management = false, the market type value will be -1)

# DSS on Run