**Modules architecture for sequential invest project**

**===============================================================================**

**Module**: import.r

**Owner**: Oz

**Input**: web download tase/yahoo/google daily prices (with dividends adjust)

**Functionality**: Download to DataRaw from yahoo/tase 10 maof stocks from 1.1.2006 till 1.1.2016 with daily prices. Choose survivor stocks. Calculate daily returns. Save dataframe as Rdata/feather to DataWork.

**Output**: dataframe StockPrices columns (Date, Stock1..10x(Price,DailyRet)), rows n=10\*12\*30

**===============================================================================**

**Module**: histogram.r

**Owner**: Oz

**Input**: StockPrices dataframe, k parameter, DDate (last day of calculation)

**Functionality**: split StockPrices to k-days sequential fragments (non-overlapping or rolling window?) with price relatives as data.

**Output**: vector of indexes Kvec to StockPrices dataframe with start of i-th fragment. Dimensions: 1x(n/k)

**===============================================================================**

**Module**: classification.r

**Owner**: Oz

**Input**: StockPrices dataframe, vector of indexes Kvec, l parameter

**Functionality**:

**Output**: dataframe Segments columns (Kstart, Lclass)

**===============================================================================**

**Module**: match.r

**Owner**: Claudiu

**Input**: StockPrices dataframe, dataframe Classifier

**Functionality**: Find best match of last k days of trading to Lclass classification (K\_Means,Kernel,RandomForest,Monkey). Metrics – Euclidean distance, weighted by how old the data is?

**Output**: BestClass - Lclass identifier best matching segment under consideration. Number 1:l

**===============================================================================**

**Module**: strategy.r

**Owner**: Claudiu

**Input**: BestClass, StockPrices dataframe, Classifier.Rdata

**Functionality**: Predict best allocation of capital among the 10 stocks based on the Lclass chosen history. Use maximum likelihood to find vector of allocation that brings to maximum compounded return. . For each method Kmeans/Kernel/RandomForest/Monkey.

Maximize

**Output**: x vector for allocation among 10 stocks for next day

**===============================================================================**

**Module**: weighting.r

**Owner**: Oz

**Input**: DDate

**Functionality**: Call histogram and classification and strategy with different k/l values and weight the different expert choices based on their compounded return.

**Output**: x allocation weighted according to success of history

**===============================================================================**

**Module**: trading.r

**Owner**: Claudiu

**Input**: StockPrices dataframe

**Functionality**: Trade chosen strategy as a backtest for all Price dataframe starting from 2 years using DDay parameter to define last date to use. Produce graphs. Calls weighting.r that calls histogram, classification and strategy files.

**Output**: Daily return dataframe DailyRet, Total return StratRet, Volatility StratStd, StratSharpe. Graphs.

**===============================================================================**

To pass data structures:

Save(var,file=”XXXX.Rdata”)

Load(“XXXX.Rdata”)