***“Signal\_Pool.py*” – Documentation**

# Description:

This file contains a Generate\_Signal class that performs the function of taking in price and Factor information to generate Technical trading signal as well as machine learning based trading signal.

# Dependencies:

* pandas
* numpy
* copy
* sklearn

# Class Function Descriptions:

|  |  |  |  |
| --- | --- | --- | --- |
| Function Name | Inputs | Output | Description |
| Constructor | Price\_Data (pd.df),  Factor\_Pool (pd.df) | None | Signal Generator is constructed by simply recording Price and Factor info and create an empty output signal pool |
| MACD | Short (int),  Long (int),  Signal\_period (int) | None | Standard MACD signal generating process. I won’t described in much details here other than just saying when MACD is positive and cross signal line downwards it’s a selling signal and vice versa. Then this signal is recorded in Signal Pool |
| Bollinger\_Band | MA\_period (int),  Band\_Width (int) | None | Standard Bollinger Band signal generating process. Short when crossing the upper bound downward and vice versa.  Same as MACD signal is added to the pool |
| RSI | MA\_period (int),  Low (int),  High (int) | None | Standard RSI generating process, short when RSI > High and buy when RSI < Low, default low and high are 30 and 70.  Signal added to the pool |
| EMA | Fast (int),  Slow (int) | None | Long when fast EMA cross Slow EMA upward and short when it crosses downward.  Signal added to the pool |
| Accum\_Dist | Period (int),  Threshold (double) | None | Calculate the accumulated distribution indicator. Then turn it into trading signal by creating a Benchmark as rolling window standard deviation of AD, then we long if the daily change of AD is greater than threshold \* Benchmark, and short if AD is smaller than (-1)\*threshold\*Benchmark |
| Machine\_Learning\_  Models | Buffer (pd.df),  Refit (int),  Use\_length (int),  Look\_Forward (int),  Feature\_period (int) | None | This function trains several machine learning models to predict the forward looking return (of Look\_Forward days). The models are by default trained on historical price and volume information of the all assets lagged (Buffer). Features are also enriched by aggregating these historical returns/ volumes by (Feature\_period). Feature enrichment is done by calling the \_Get\_ML\_Features function. However, these factors can be overwritten by the factors provided in class construction. All models are refitted every (Refit) days using (Use\_length) number of observations every refit.  Currently I included 5 Machine learning models including: LASSO, Elastic Net, Neural Network, Support Vector Machine and Random Forest |
| Generate\_All\_Signals | Buffer (pd.df)  Refit\_freq (int) | pd.df | This is a wrapper function of all above signals and it returns the signal pool that contains all 10 above stated signals. Both arguments are passed along to generate the ML signals |
| \_Get\_ML\_Features | Period (int)  Buffer (pd.df)  Feature\_time (int) | Pd.df | This is an internal function that takes care of turning factors/historical prices/volumes into the features used in the machine learning signal generation. |