**Design Doc:**

1. **Folder Structure:**

* [**finproduct**](../../finProduct):
  + documents
  + [utility](#utility)

* + [stockprediction](#_stockprediction)
  + optionvaluecalculation
  + optionstrategy
  + directtrading
  + portfoliooptimizer
  + reporting

## **utility**

* + database
    - <features.csv> 🡪 *contains input features like MA,RSI etc*.
    - <user_ip.csv> 🡪 *contains user input args for historic data*.
    - dataframe.csv 🡪*dump pandas dataframe using pickle.*
    - *{continuously dataset.}*
  + class GetLoad\_Uip(object)
    - def GetIp(self): 🡪 *function to read user input and save in database.*
    - def GetLoad\_FeatureIp(self): 🡪 *function to read feature input from user and save in database.*
    - def LoadData(self,csv\_name) : *function to read data from csv(tables) and return same.*
  + logs 🡪 *logs for utility folder mainly for GetLoad\_Uip class*

## **Stockprediction**

* + Technical\_Analysys
    - class technical\_PD(object):
      1. read data from csv , convert it to dictionary and pass it to nsepy.get\_history . [Note : if there is change in arguments of nsepy.get\_history then code change will be required .]
      2. create a data frame using pandas and save it to dataframe.csv
      3. Calculation of technical indicators :
         * Read the technical indicators from file.

Format should be like :

**MA**,**5**,10,

**15  
RSI**,**5**,**10**,**15**

**We will calculate technical indicators as follows:**

|  |
| --- |
|  |
|  | MA\_day = [5,10,15] |
|  | for ma in MA\_day: |
|  | column\_name = 'MA for %s days' %(str(ma)) |
|  | dataset[column\_name]= pd.rolling\_mean(dataset['Close'],ma) |
|  |  |

Calculation of ROI

dataset ['Daily Return'] = dataset ['Close'].pct\_change()

* + logs

**Questions:**

1. ROI is predicted value /predicted value – todays value.