```
        else:
            print('Unidentified error !\n')

        ws.Beep(1000,1000)

    if  file_name != '':

        interact(choose_target_features, target = options, feature=options.insert(0,'- All features -'), confirm=['No','Yes'])
```

```
        MACHINE LEARNING - LINEAR REGRESSION MODEL BUILDER

        Informe path/name of CSV file:  (<ENTER> to abort)
        C:/Users/p_com/Downloads/auto_prices.csv

        Opening file ...
        Done !

                target  | price                 ▽ |
                feature | - All features -      ▽ |
                confirm | Yes                   ▽ |
```

Features identified on file that present a linear tendency with the target   PRICE   (correlation coefficient >= 0.7):

  -   width   (0.751265)
  -   city-L/100km   (0.789898)
  -   horsepower   (0.809575)
  -   curb-weight   (0.834415)
  -   engine-size   (0.872335)

Using 75 % percentage of data to build up model

------- Checking relevant features for an order-1 model...

Grid-Searching them with 2 fold(s) ...
Grid-Searching them with 3 fold(s) ...
Grid-Searching them with 4 fold(s) ...
Grid-Searching them with 5 fold(s) ...

 ----- Results ------> Most relevant futures so far: ['city-L/100km', 'curb-weight', 'engine-size']

                    Best order so far............:   1

                    Best cross-validation........:   4   fold(s)

                    Score for above..............:   0.7874275354801352

------- Checking relevant features for an order-2 model...

Grid-Searching them with 2 fold(s) ...
Grid-Searching them with 3 fold(s) ...
Grid-Searching them with 4 fold(s) ...
Grid-Searching them with 5 fold(s) ...

 ----- Results ------> Most relevant futures so far: ['width', 'city-L/100km', 'engine-size']

                    Best order so far............:   2

Best cross-validation........: 4 fold(s)

                    Score for above..............: 0.805583701586542

 ------- Checking relevant features for an order-3 model...

Grid-Searching them with 2 fold(s) ...
Grid-Searching them with 3 fold(s) ...
Grid-Searching them with 4 fold(s) ...
Grid-Searching them with 5 fold(s) ...

 ----- Results ------> Most relevant futures so far: ['city-L/100km', 'engine-size']

                    Best order so far............: 3

                    Best cross-validation........: 4 fold(s)

                    Score for above..............: 0.8112658971389225

 ------- Checking relevant features for an order-4 model...

Grid-Searching them with 2 fold(s) ...
Grid-Searching them with 3 fold(s) ...
Grid-Searching them with 4 fold(s) ...
Grid-Searching them with 5 fold(s) ...

 ----- Results ------> Most relevant futures so far: ['city-L/100km', 'engine-size']

                    Best order so far............: 4

                    Best cross-validation........: 4 fold(s)

                    Score for above..............: 0.818295462192407

 ------- Checking relevant features for an order-5 model...

Grid-Searching them with 2 fold(s) ...
Grid-Searching them with 3 fold(s) ...
Grid-Searching them with 4 fold(s) ...
Grid-Searching them with 5 fold(s) ...

 ----- Results ------> Most relevant futures so far: ['city-L/100km', 'engine-size']

                    Best order so far............: 4

                    Best cross-validation........: 4 fold(s)

                    Score for above..............: 0.818295462192407

 Using Ridge Regression to improve model with the parameters above ...

 Grid-Searching with alpha: 1e-06

 Grid-Searching with alpha: 0.26214499999999996

 Grid-Searching with alpha: 0.393217

 Grid-Searching with alpha: 0.409601

 Grid-Searching with alpha: 0.41779299999999997

 Grid-Searching with alpha: 0.41804899999999995

 Grid-Searching with alpha: 0.41817699999999997

 Grid-Searching with alpha: 0.418193

 New score obtained: 0.822600 (0.53 % improvement)

********* FINAL RESULTS ***********

Polynomial order.................: 4
Bias included (alpha)...........: 0.418192
Predicted Price.................: Model constituted on 75.00 % of Training Data (Cross Validation with 4 parts)
R2 on Testing Data 25.00 % .....: 0.82522452676911 (Cross Validation with 4 parts)

R2 on Training Data 100 % ......: 0.8583293519085275
Model on Training Data 100 % ...: y-hat = +9702.193461
                                          -2319.102563*((city-L/100km))
                                          -146.728110*((engine-size))
                                          -3723.461951*((city-L/100km)^2)
                                          +646.810815*((city-L/100km) (engine-size))
                                          -21.675050*((engine-size)^2)
                                          +708.758145*((city-L/100km)^3)
                                          -101.503763*((city-L/100km)^2 (engine-size))
                                          +2.207947*((city-L/100km) (engine-size)^2)
                                          +0.066966*((engine-size)^3)
                                          -31.217227*((city-L/100km)^4)
                                          +4.094067*((city-L/100km)^3 (engine-size))
                                          -0.044853*((city-L/100km)^2 (engine-size)^2)
                                          -0.005549*((city-L/100km) (engine-size)^3)
                                          +0.000033*((engine-size)^4)

Population Distribution on price