1. i) The absolute difference between them is 41504.63259835

   ii) We know that (gradient * Learning_rate) becomes smaller and smaller as we approach the minima of any convex function, so I took a very small value ‘tolerance = 0.001’ as a flag to know that I have reached the minima of the function.
   Therefore, Stopping criterion is to break when
   gradient * Learning_rate<= tolerance

   Also I checked the weights with their closed form solution to insure that optimum iterations and learning rate is set for all methods (SGD,GD,P-norm with L2 and L4 regularizer)

   iii) The absolute difference between them is 13587.79588679

2. I optimized p-norm regularized objective function (where p is 2 or 4) using gradient descent

   Lambda value I took for both is 1000

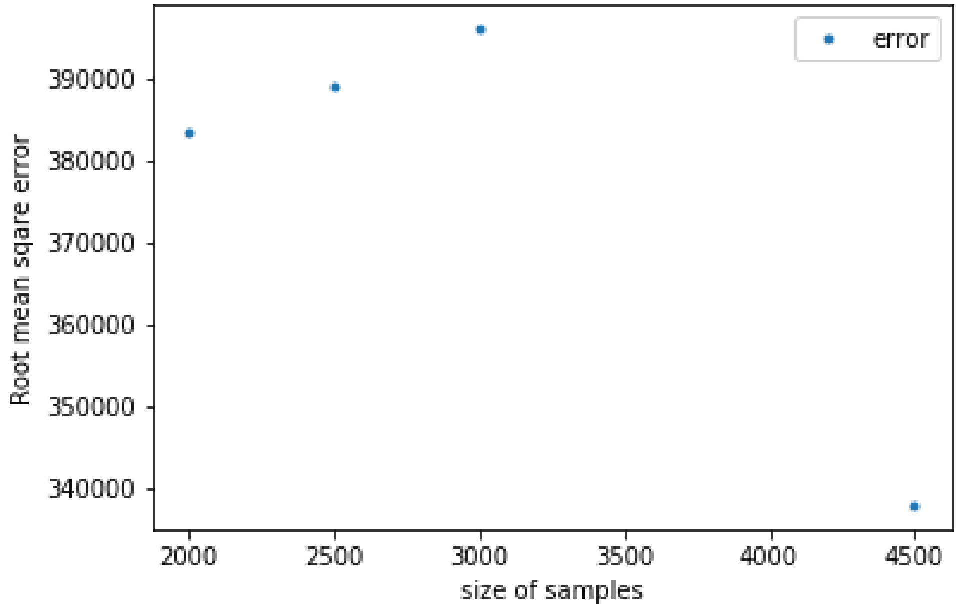
   Root mean squared error when p=2 is 425279.35097577

   Root mean squared error when p=4 is 694515.74717032

3. I chose a polynomial basis function for which the error on train and dev set was reduced. It was mostly hit and trial type where I choose that power for which error reaches its minimum like -
   engine^0.75 , year^5 , torque^3 , seats^1.5 , etc

   Root mean squared error on development set samples(dev.csv) is 413468.41950515

4. Here's the plot which I got -



5. The two least useful ones are fuel and mileage.

   It is so because when I calculated the correlation between fuel & selling price and mileage and selling price, that came out to be very less (-0.0535159159054682) and (-0.1096452342870396)

   Correlation value lies between 0 and 1 where value of 1 means both data sets are in strong linear relationship whereas value of 0 means no relationship exists between compared quantities.

6. In this part, first I looked into the graphs between different features vs selling price and also looked into correlation between them. These things helped me to better predict the basis function which i can use to make their correlation high ,i.e. stronger linear relationship

Then,I used some polynomial basis and tried to get less error than what I get from the data set without basis. Luckily I got slightly lesser error with basis for some entries as you can see by my submissions on kaggle :D (not too much difference though)

Normalising the data sets helped in quicker processing and results.
I also used the polynomial basis functions which I used in part 3 of Q , though some had to be dropped as they increased error than decreasing :(