University of Costa Rica

PF-3115: Computational and statistical techniques of Machine Learning

Lab report #1

Osvaldo Ureña A55783

María José Cubero B22148

1. Exploratory data Analysis.

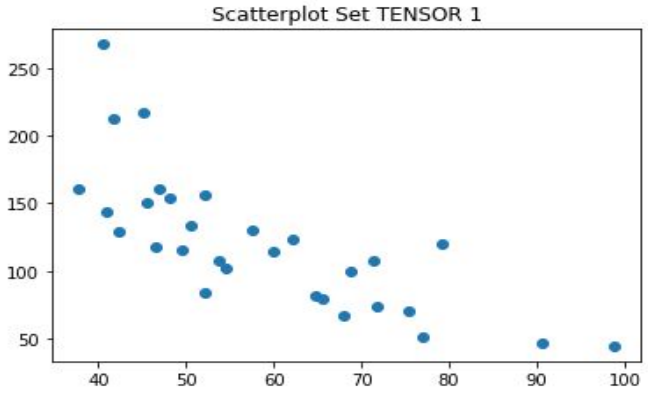
In the following comparative square we can see two important things, first of all the descriptive statistics of the two variables ActionLatency and APM:

	ActionLatency	APM
Maximun	176.37	389.83
Minimun	24.09	22.06
Mean	63.74	117.05
Standard Deviation	19.24	51.95

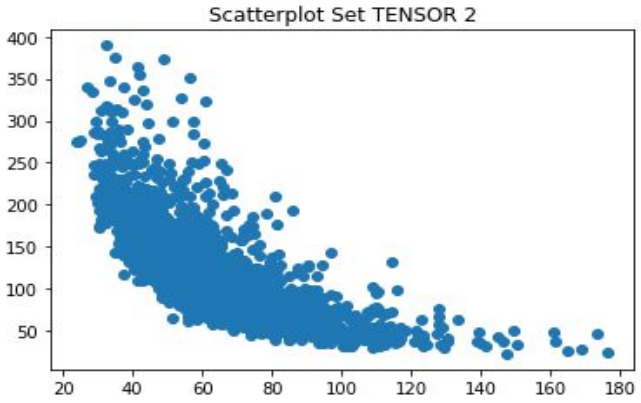
It tells us that APM has a little more variability than Action Latency, and also the range of APM is greater. The correlation between the two variables is -0.72; since we want to predict APM, the correlation of -0.72 implies that when ActionLatency increases then APM decreases.

The data is splitted into three data sets, data set 1 with 30 observations and then a training set with 80% of the data and a test set with 20% of the data.

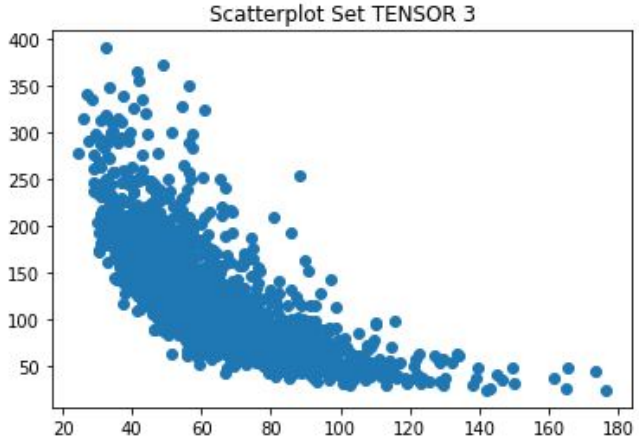
Here is the graph with the 30 first observations:



Here is the graph with 80% of the data:



And the next is the graph with 20% of the data.



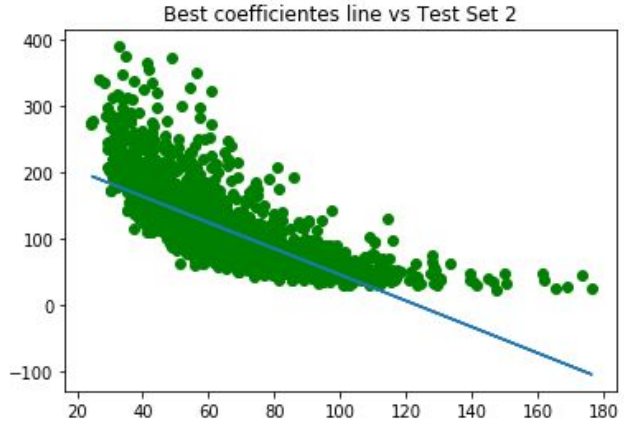
we can see in the three graphs the negative correlation between the variables.

2. Ordinary Least Squares

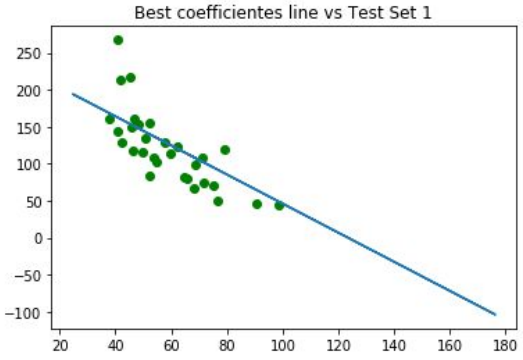
We fit a line with the ordinary least squares method, we used the training set to fit the line. The training set and the test set change everytime we run the lab, at the time we ran the lab the line was: $y = -1.97 * x + 242.76$. We graph this line in to the three graphs, here is the result:

a. Best coefficients line into the scatterplot of the training set.

b. Best coefficients line in to the test set.



c. Best coefficients line in to the 30 first observations set.



3. Fitting a linear model with gradient descent

In this part we calculated the *w* and *b* by the method of the gradient descent, in order to do that we need another parameter "*alpha*". At the beginning we didn't know the values of *w* and *b*, we started with *w*= 0 and *b*=1 but something interesting happened, if we set an alpha =1 then *w* blew up and the loss was 'na'. If we set an *alpha* = 0.00001 then w sets an acceptable value, but there is something wrong with *b* because it lasts so much to set an acceptable value.

It is a problem because we didn't know the correct values for w, b and alpha. To solve the problem we did a lot of iterations and set a lot of values for *w*, *b* and *alpha*, and at last we noticed that if we set a value of b higher than 250 the value of *b* goes down and if we set a value of *b* lower than 200 the value of *b* goes up, also we have a clue of the values of *w* and *b* because the OLS tells possible values for *w* and *b*. We ran the model with the following values:

w= -1
b= 225
alpha= 0.000001
n= 20000

The result was the following linear equation: *y*= *x** -1.717141 + 224.991302 with a loss value of 1277. To solve the problem with *w* and *b* we also calculated a new x, x=Xn, where Xn is *(x-mean(x))/(range(x)/2)* that new input (Xn) fit the following line: *y*= *x** -1.371933+220.727539.

We used three methods to do the training: manually, auto and optimized. First, we calculated the descent gradient manually (training function in the lab), then we calculated the descent gradient with the help of Pytorch (training_auto function), and then we calculated the descent gradient by the method SGD of Pytorch (trainig_opt_SGD function) and also with the method Adam of Pytorch (trainig_opt_ADAM) .

The results we got when we ran all of these implementations are practically the same, as you can see in the table below:

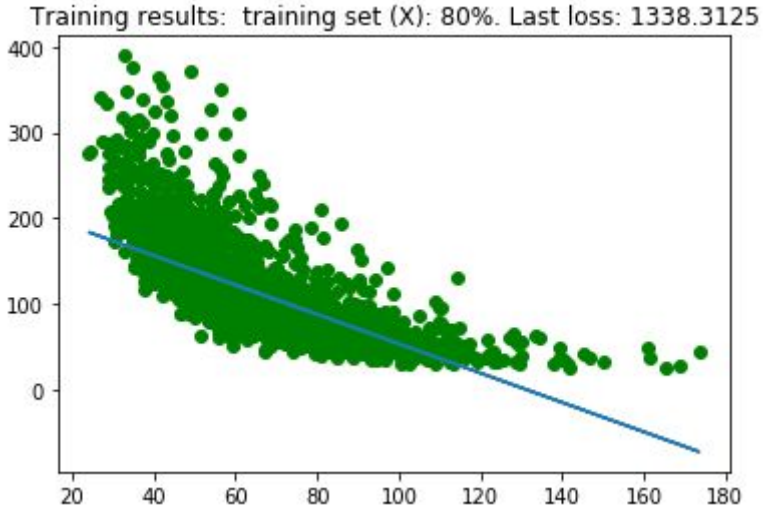
	w	b	loss
Manually	-1.379751	220.727539	13411.572266
Auto	-1.386032	220.727448	13411.4355
Opt (SGD)	-1.379751	220.727539	13411.5723
Opt (Adam)	-1.398824	220.72753	13411.1680

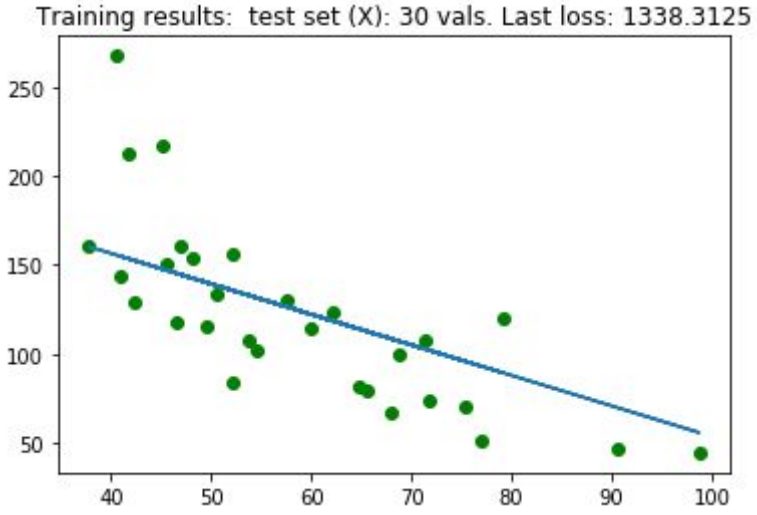
4. A better fit

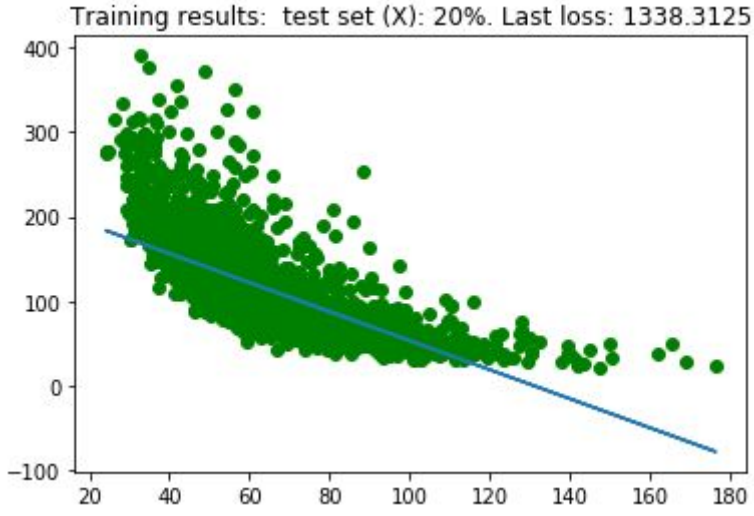
For this part of the laboratory, we tried to fit a non-linear model but the outcome wasn't the expected, due to the fact that we couldn't adjust the parameters to get the correct results. We tried to do it with 200000 iterations and tried to adjust it with different values. To see the result that we got take a look at the section g of the Anexxes.

Annexes

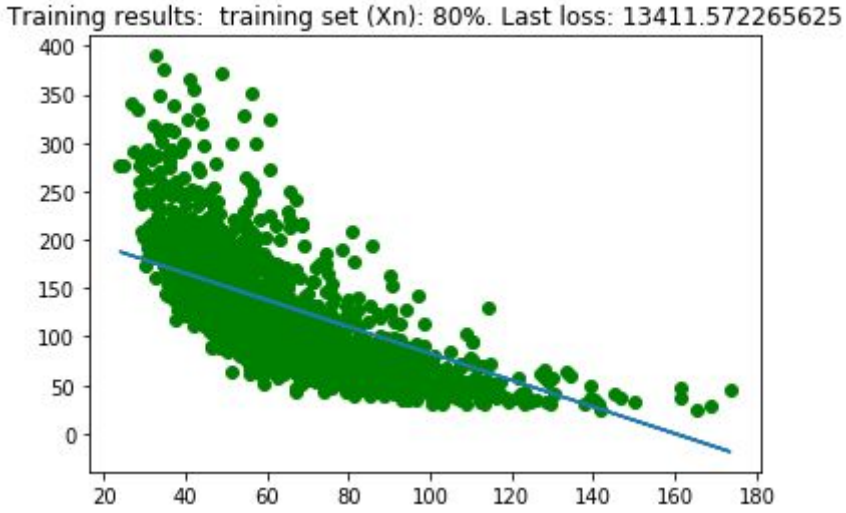
a. Training results with X not normalized

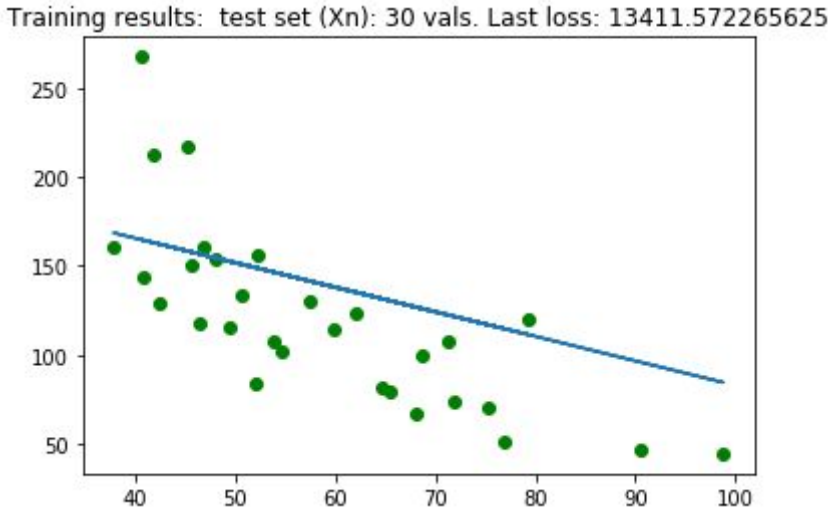


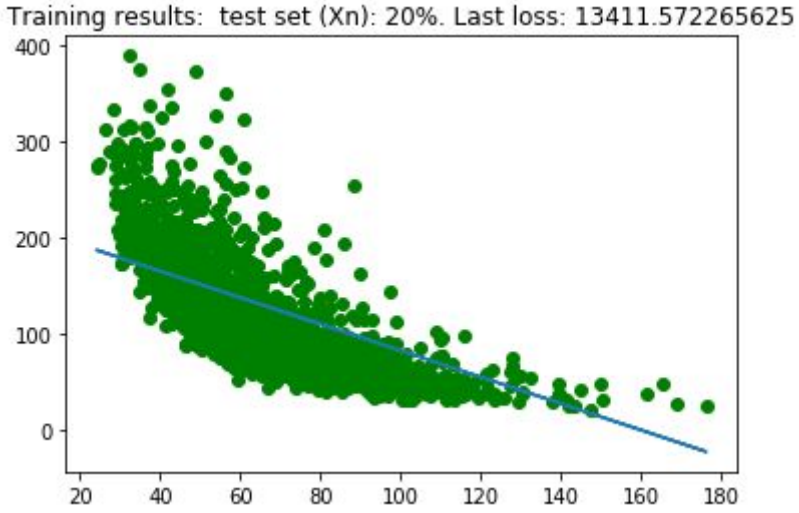




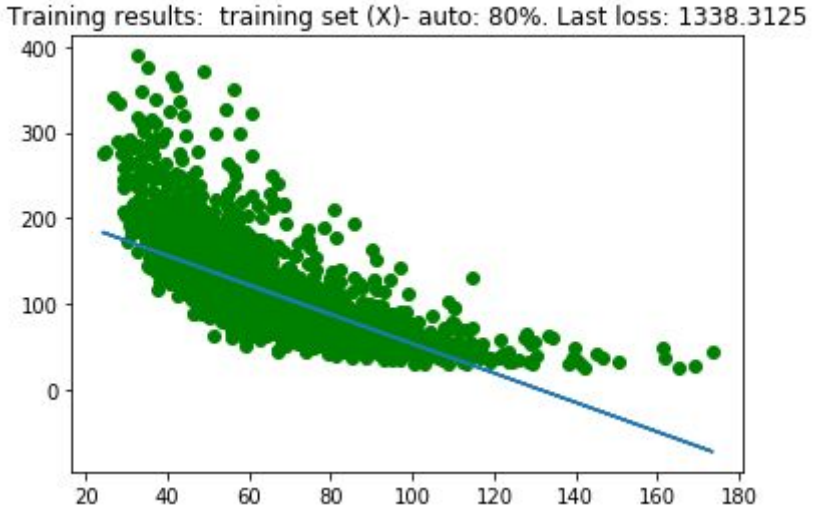
b. Training results with Xn normalized

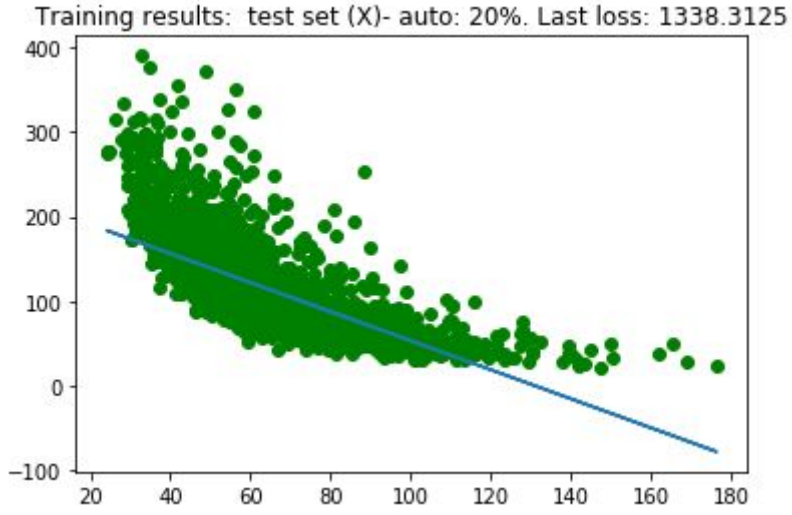




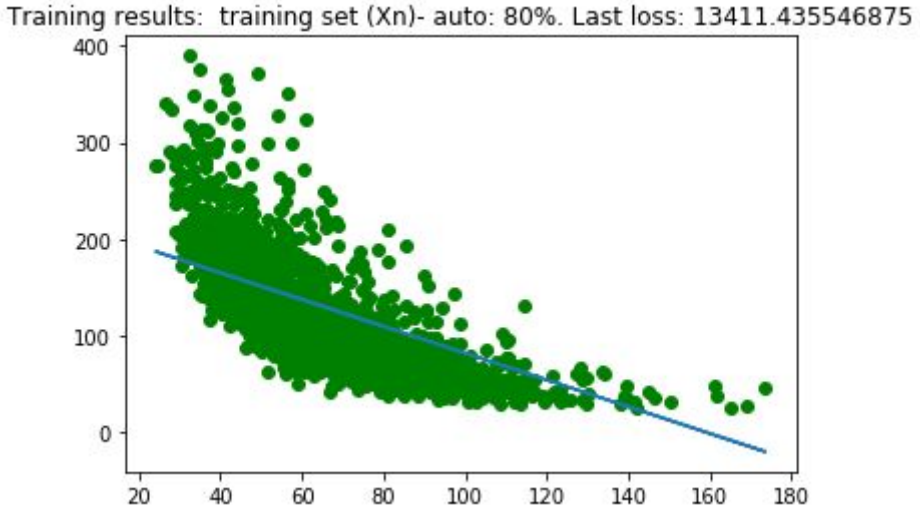


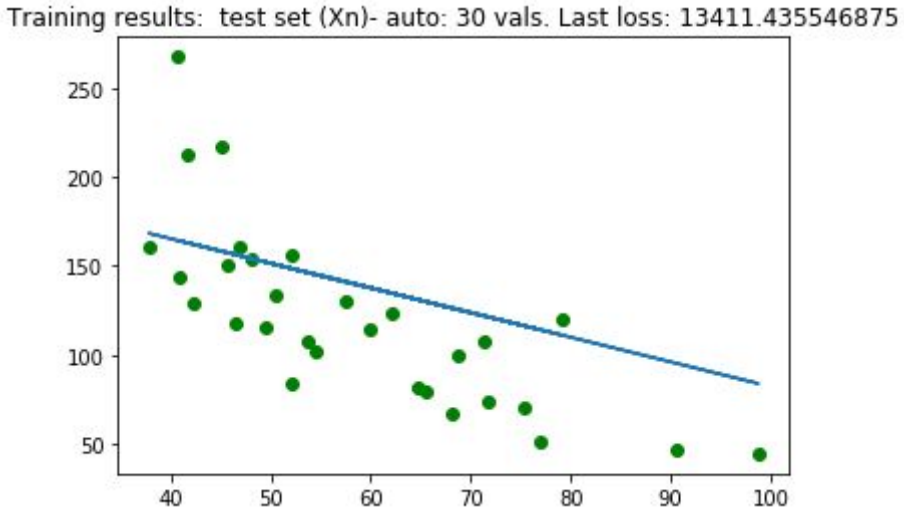
c. Training auto with X not normalized

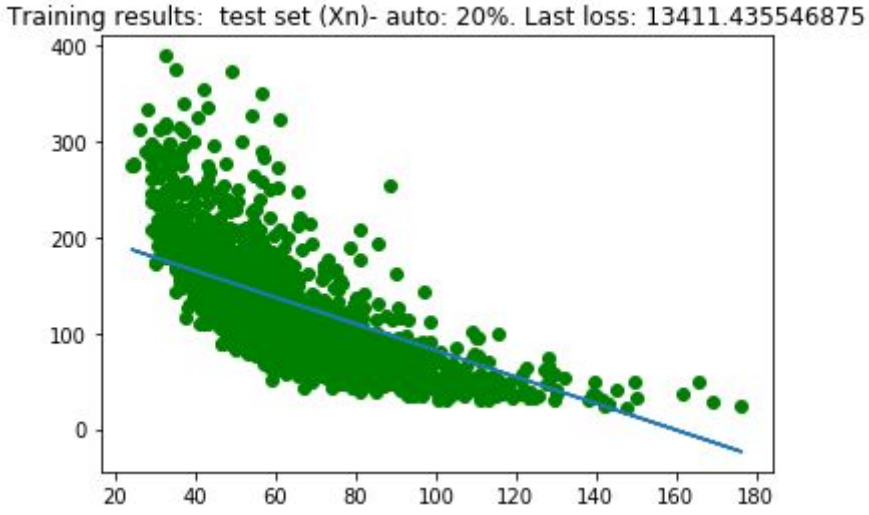




d. Training auto results with X normalized

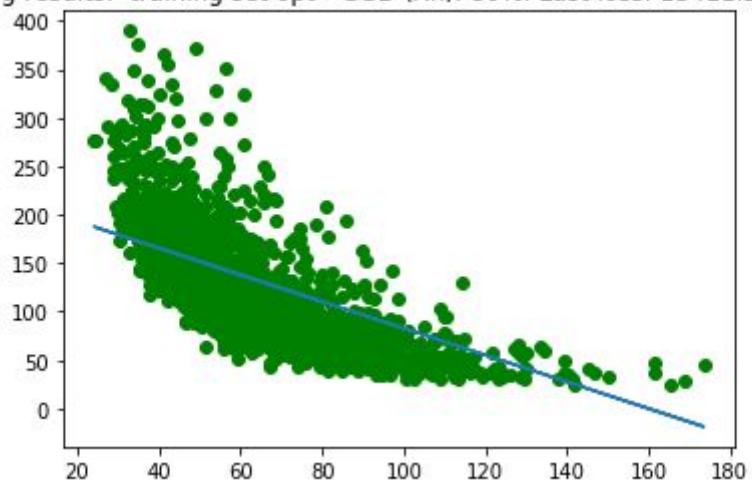




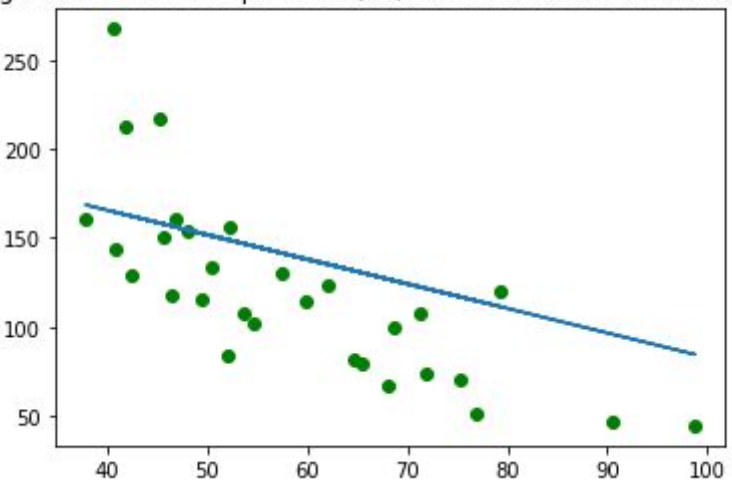


e. Training opt results using SGD with X normalized

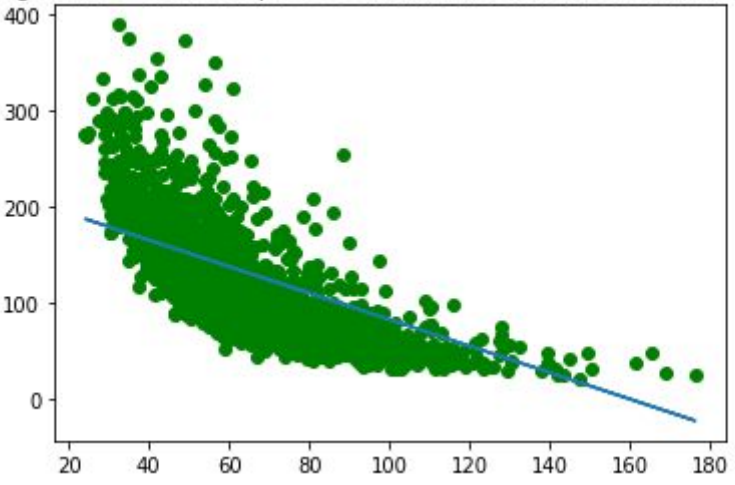
Training results: training set opt->SGD (Xn): 80%. Last loss: 13411.572265625



Training results: test set opt->SGD (Xn): 30 vals. Last loss: 13411.572265625

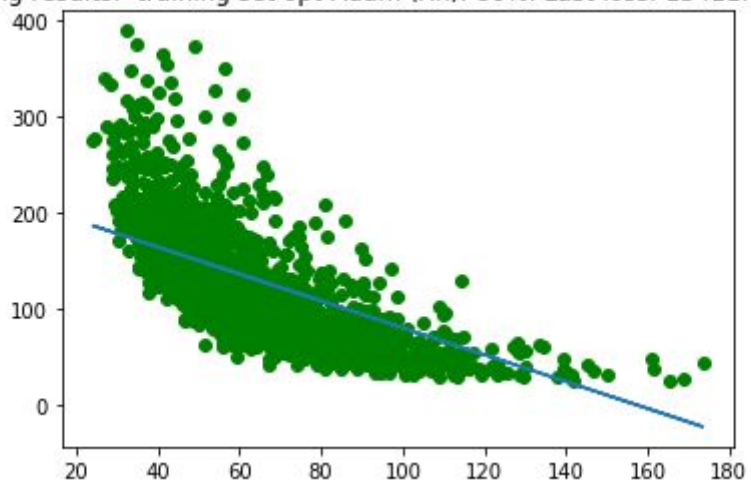


Training results: test set opt->SGD (Xn): 20%. Last loss: 13411.572265625

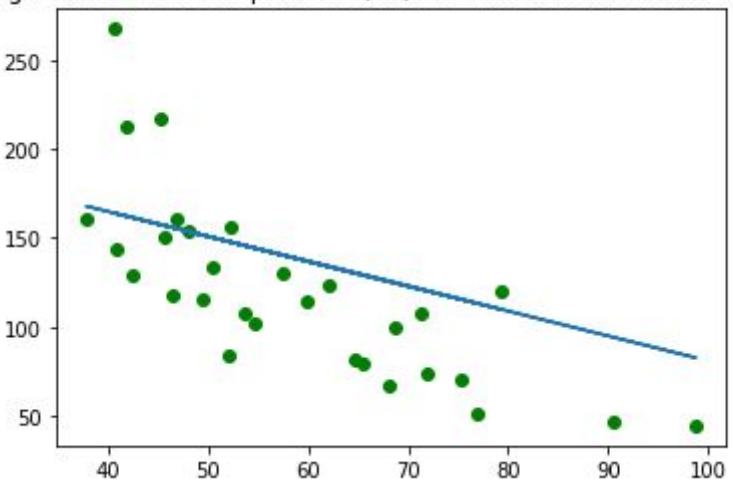


f. Training opt results using SGD with X normalized

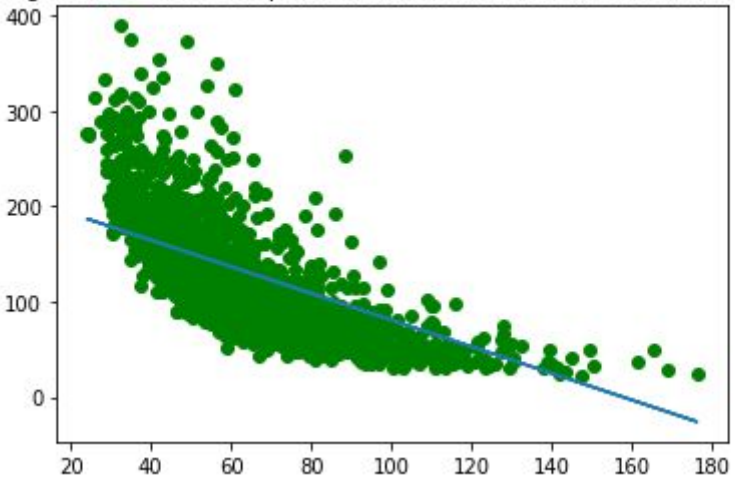
Training results: training set opt-Adam (Xn): 80%. Last loss: 13411.16796875



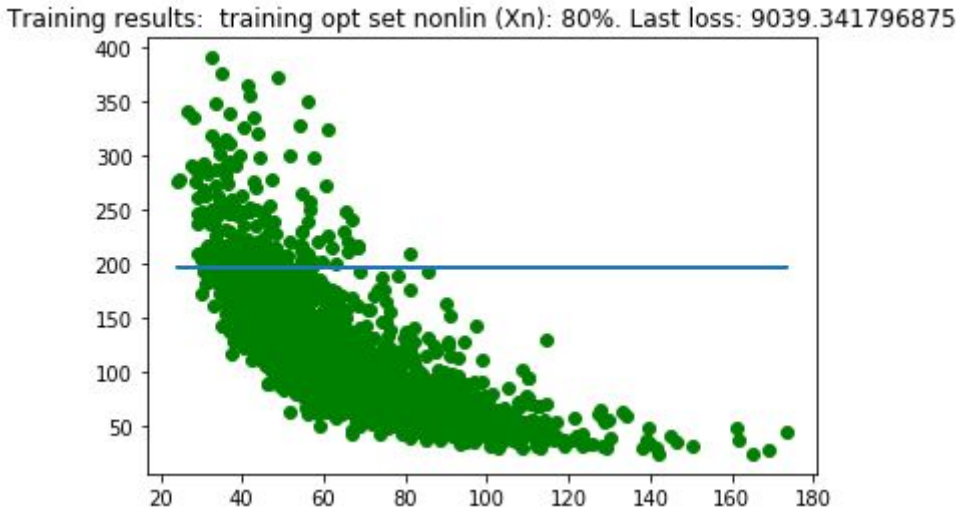
Training results: test set opt-Adam (Xn): 30 vals. Last loss: 13411.16796875

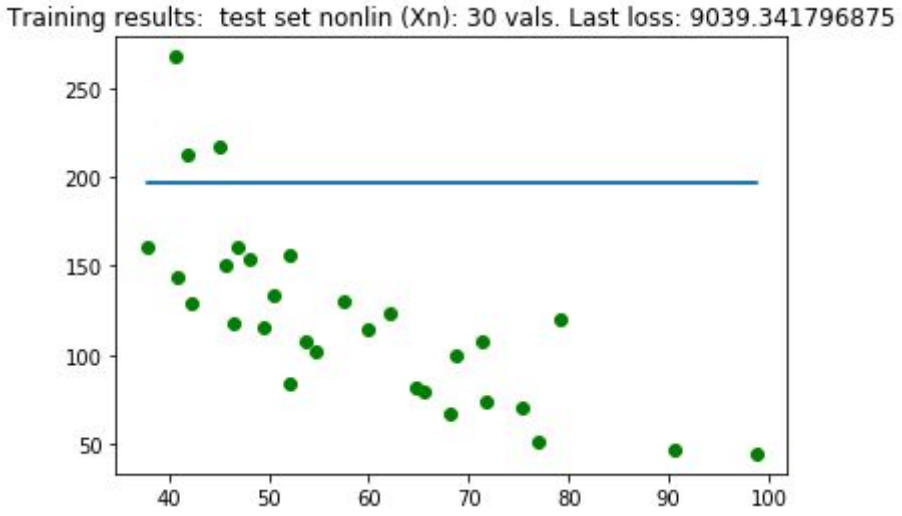


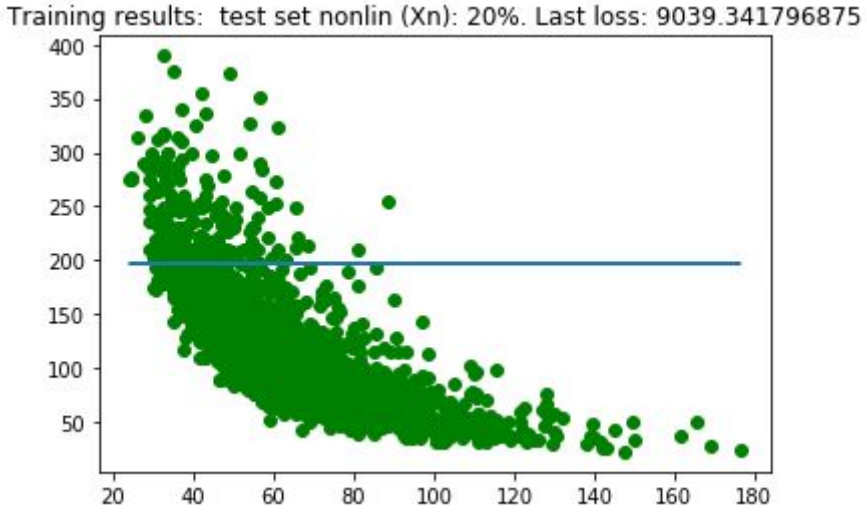
Training results: test set opt-Adam (Xn): 20%. Last loss: 13411.16796875



g. Training opt results using the non-linear function with X normalized







h. Exploratory data analysis

```
1. Exploratory Data Analysis**********************************************************************

---ActionLatency---:
 Maximun: 176.37
 Minimum: 24.09
 Mean: 63.74
 Standard deviation: 19.24

---APM---:
 Maximun: 389.83
 Minimum: 22.06
 Mean: 117.05
 Standard deviation: 51.95

Correlation between ActionLatency and APM :-0.72
```