Project part 2

- **Initial attempts:**

  during our initial attempts we had many problems such as:

  - Since we deal with big numbers our loss function and features weights got so big that it became nan after few iterations since the learning rate was not small enough.
  - In other attempt with the neural network we had a dying ReLU. All the weights became zero and the loss function did not change even after thousands of learning loops. A dying ReLU always outputs the same value for any input. Probably it happened because we had large negative weights.

- **How we improved our initial attempts?**
  - After trying with an arbitrary learning rate and getting infinity results for the loss function time after time, we decreased the learning rate to be 0.0000001.
  - Since the learning rate is very small, we increased the number of learning iterations to 100,000.
  - We normalized our data to smaller numbers so we could increase the learning rate and get good results faster.
  - After normalizing the data, the dying ReLU problem was solved.
  - We tried adding some new features which computed as follows:
    for every important feature X we added X^2 as a new features.
    In total we added 10 new features.
  - We set the hidden layers number to be one.
  - We set the number of neurons in the hidden layer to be 150 after trying more and less neurons.

- **Failed attempts to improve the model**
  - We tried to change the activation function to be SeLU or ELU, but it only made the results worst.
  - We tried to add another hidden layer but it did not improve the results, and only slower our model.
  - We tried to change the number of neurons in the hidden layer and it worsen our results.
  - In the linear regression model if we tried to normalize the data or to add features the weights became nan after few iterations.

- We tried to increase the test data precent and we got very interesting results. The loss function and train error decreased to 234,596,110 but had big jumps from one iteration to another, and the test error increased to 320,649,200

```
Iteration: 90000  loss: 235168100.0
Iteration: 91000  loss: 234616320.0
Iteration: 92000  loss: 234084300.0
Iteration: 93000  loss: 233591940.0
Iteration: 94000  loss: 235084910.0
Iteration: 95000  loss: 235276020.0
Iteration: 96000  loss: 234652880.0
Iteration: 97000  loss: 233160270.0
Iteration: 98000  loss: 233551470.0
Iteration: 99000  loss: 232914270.0
Iteration: 100000  loss: 234596110.0

Test error : 320649200.0
```

- **Results:**
  - **Average check** - In order to know if our model is better than the average prediction we calculated the average and compared it to the real data.
    We calculated the variance of the train and the test.
    In the screen shot you can see that with the average prediction we get in the neural network variance of 11 billion. It is easy to see that our network is much more accurate.

```
#what will happend if we will predict the average

#check for avg train
avg_train=sum(new_y_train)/len(new_y_train)
print("avg_train:",avg_train)
#check for var train in compare to loss function
var_train=0
for i in range(0,len(new_y_train)):
    var_train=var_train+(new_y_train[i]-avg_train)**2
print("var_train:",var_train/len(new_y_train))
#check for var test in compare to test error
var_test=0
for i in range(0,len(new_y_test)):
    var_test=var_test+(new_y_test[i]-avg_train)**2
print("var_test:",var_test/len(new_y_test))
```

```
avg_train: 222483.62697065243
var_train: 11425199834.210613
var_test: 11065594439.37833
```

- **Linear Regression-** with 100,000 iterations and learning rate = 0.0000001 we got train error=3,442,777,600
  And test error= 3,401,281,500

$(this\ means\ that\ the\ average\ prediction\ error\ is\ \sqrt{3{,}401{,}281{,}500} = 58{,}320)$

```
Iteration: 95000 Loss:  3459173000.0

Iteration: 96000 Loss:  3455799600.0

Iteration: 97000 Loss:  3452475400.0

Iteration: 98000 Loss:  3449196500.0

Iteration: 99000 Loss:  3445960700.0

Iteration: 100000 Loss:  3442777600.0

Test error 3401281500.0
```

- **Neural Network-** With 150 neurons in one hidden layers, 100,000 iterations and learning rate = 0.0000005 we got train error=240,848,770
  And test error= 315,483,840

```
Iteration: 94000  loss: 239951900.0
Iteration: 95000  loss: 240376600.0
Iteration: 96000  loss: 240627890.0
Iteration: 97000  loss: 240587790.0
Iteration: 98000  loss: 240330820.0
Iteration: 99000  loss: 240976720.0
Iteration: 100000  loss: 240848770.0

Test error : 315483840.0
```

$(this\ means\ that\ the\ average\ prediction\ error\ is\ \sqrt{315{,}483{,}840} = 17{,}761)$