Linear Regression with Batch Gradient Descent

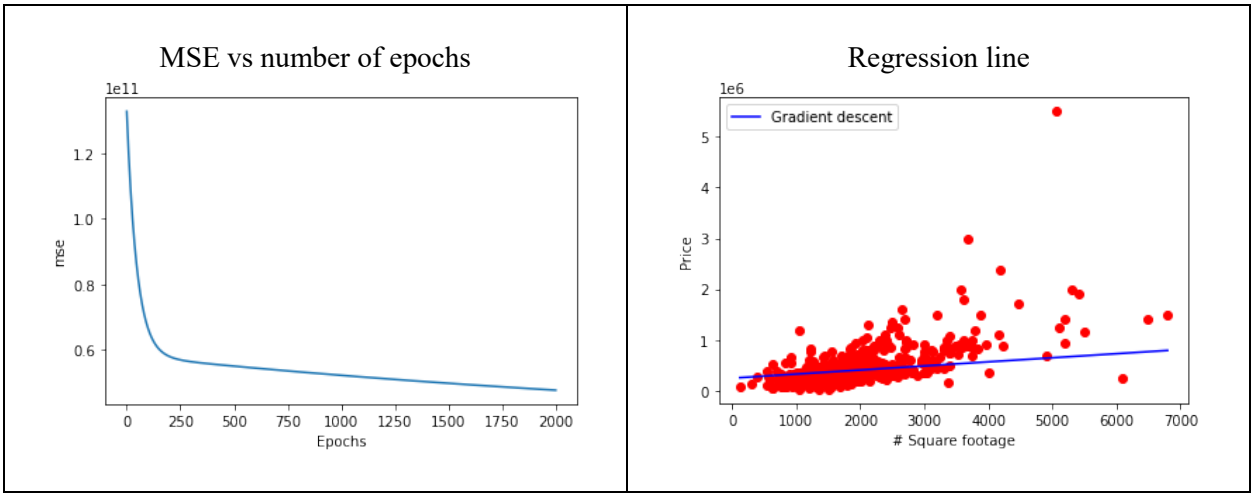
In this project you will train a linear regression model using a dataset that contains a collection of real estate listings in San Luis Obispo county. Given the square footage, the model can then predict the price of the house. The dataset contains the multiple fields:

1. MLS: Multiple listing service number for the house (unique ID).
2. Price: the most recent listing price of the house (in dollars).
3. Bedrooms: number of bedrooms.
4. Bathrooms: number of bathrooms.
5. Size: size of the house in square feet.
6. Price/SQ.ft: price of the house per square foot.

The start-up python code reads the dataset, chooses the field *size* to create a 1D training set and normalizes it with min-max scaling. Complete this code to perform the following tasks:

- Train the model with gradient descent.
  - Loop for 2000 epochs. This number is small enough to allow you to debug the code quickly. You're welcome to increase the number of epochs to get better results.
  - Initialize the parameter vector **w** with random numbers.
  - Initialize the learning rate to 0.01.
- Generate a graph of MSE as a function of the number of epochs
- Plot the regression line over the training examples. The line is defined by $\mathbf{w} = [w_0\ w_1]^T$
- Print the predicted price of a 5000 square foot house. Remember to "normalize" the square footage first.

Expected outputs:



Submit the python code (*.ipynb* file). Your code must run on Google Colab.

Discussing this project with other students is highly recommended but you have to submit your own solutions.

```
## Plot fitted curve

plt.plot(X[:,1] * max_min + Xmin, y, 'ro')
plt.plot(Xsort[:,1] *max_min + Xmin, yhat, 'b', label="Gradient descent")
plt.legend()
plt.xlabel("# Square footage")
plt.ylabel("Price")
plt.show()
```