Assignment 1

Yufei Liu

260561054

Question 1

1.1

Pseudocode:

```
pairs = [(.2, "Movies"), (.4, "COMP-551"), (.1, "Playing"), (.3, "Studying")]
probabilities = numpy.random.multinomial(n, zip(*pairs)[0])
result = zip(probabilities, zip(*pairs)[1])
```

1.2

Sample for 100 days, we get:

```
[(22, 'Movies'), (41, 'COMP-551'), (11, 'Playing'), (26, 'Studying')]
```

Sample for 1000 days, we get:

```
[(214, 'Movies'), (364, 'COMP-551'), (102, 'Playing'), (320, 'Studying')]
```

We can see that these two fractions are generally in line with the multinomial distribution. It also has some variances when sampled to large data scale. And the variances increase as the data scale increases.

Question 2

2.1

(a)

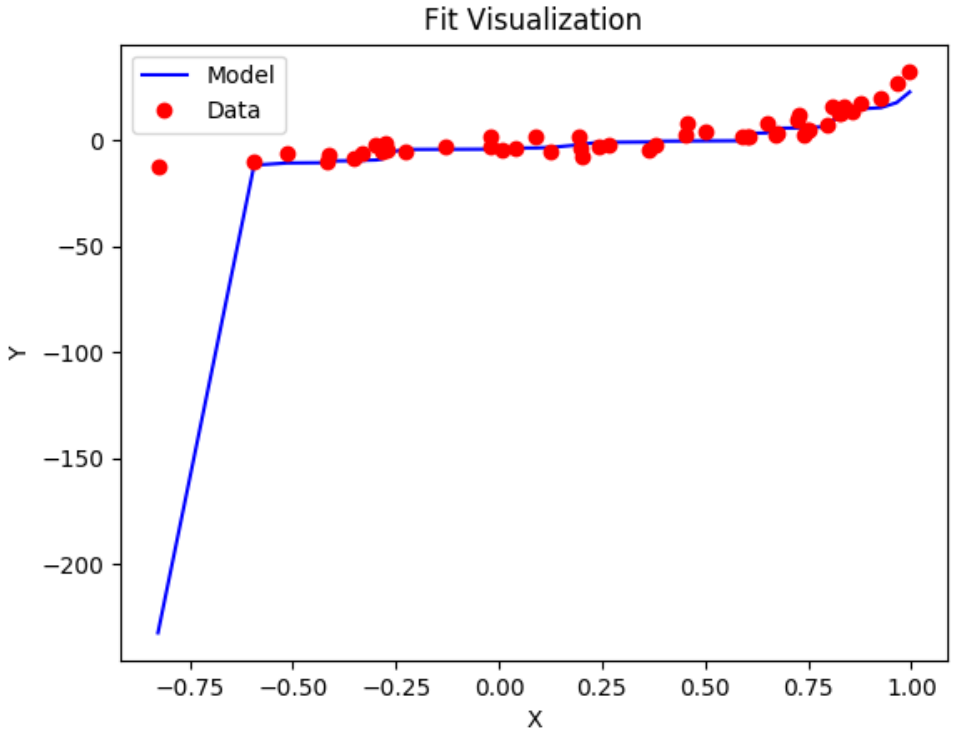
MSE for Training set (No regularization) = 6.474753981824253

MSE for Validation set (No regularization) = 1414.3778093802093

We can see that the validation MSE is really large, which means this model is not good.

(b)

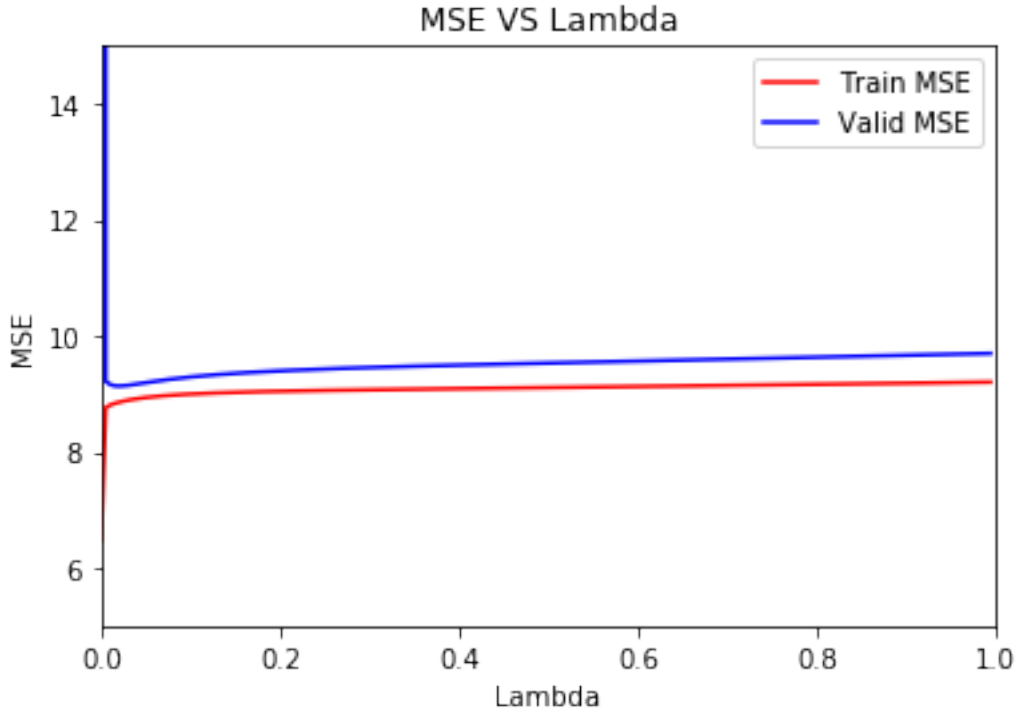
Fit visualization for validation dataset



(c) This model is overfitting because the training MSE is much smaller than validation MSE, which means the model is only a good fit for training dataset.

2.2

(a)



(b)

Minimum MSE: 9.135098784694344

Lambda: 0.02

The best value of λ is 0.02, because validation MSE is at its lowest.
The test MSE based on λ = 0.02 is 10.73

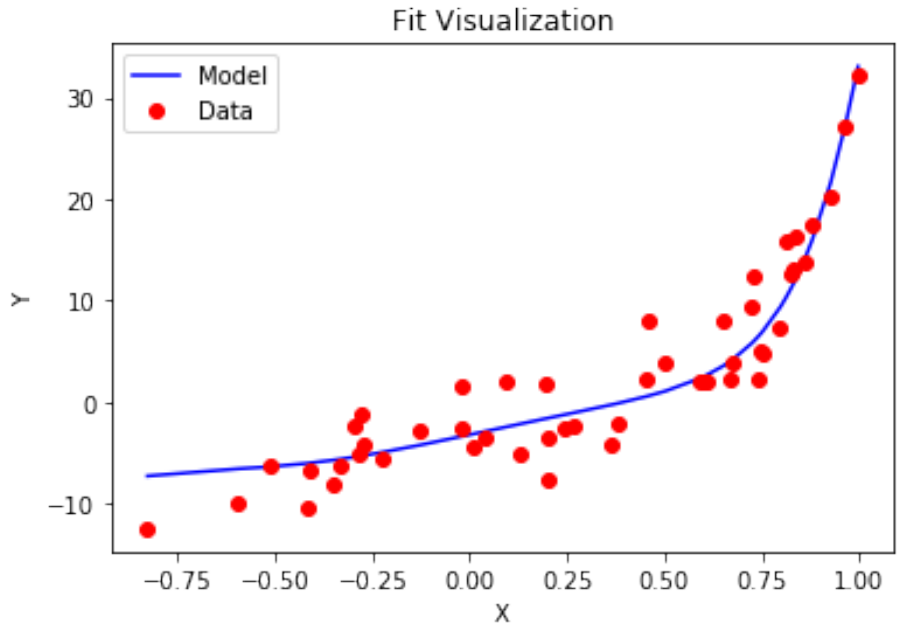
MSE of Test data set (L2 regularization): 10.730218400927383

The test performance is good because its MSE is really close to validation MSE, which meets our expectation.

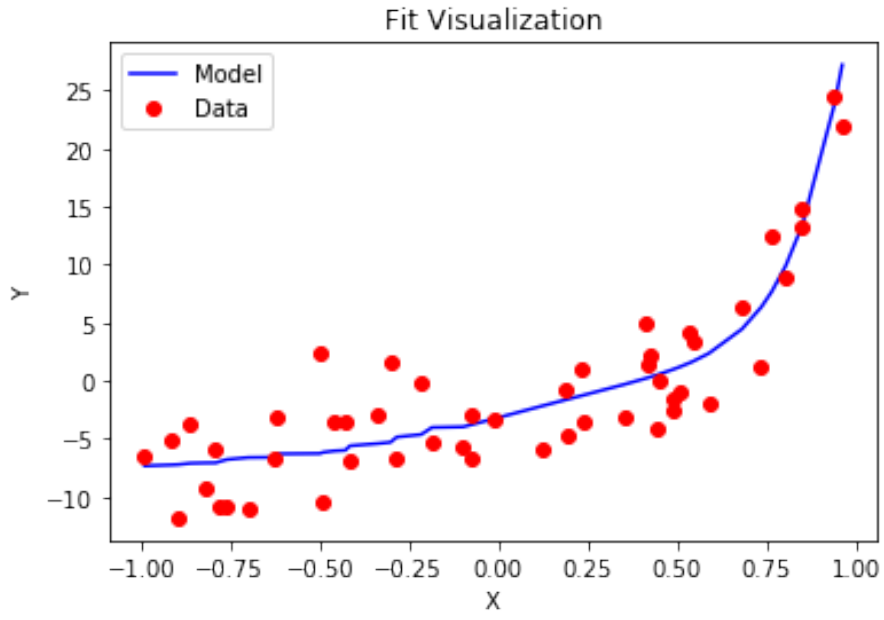
(c)

Visualize the fit of validation dataset with λ = 0.02

(See next page)



Visualize the fit of test dataset with $\lambda = 0.02$



(d)

This model is not overfitting nor underfitting, because by using the λ we choose, the MSE of validation set and test set is really close.

3.

By looking at the fit visualization and polynomial function graph, I think the curve above looks really similar to a 4-degree polynomial function curve.

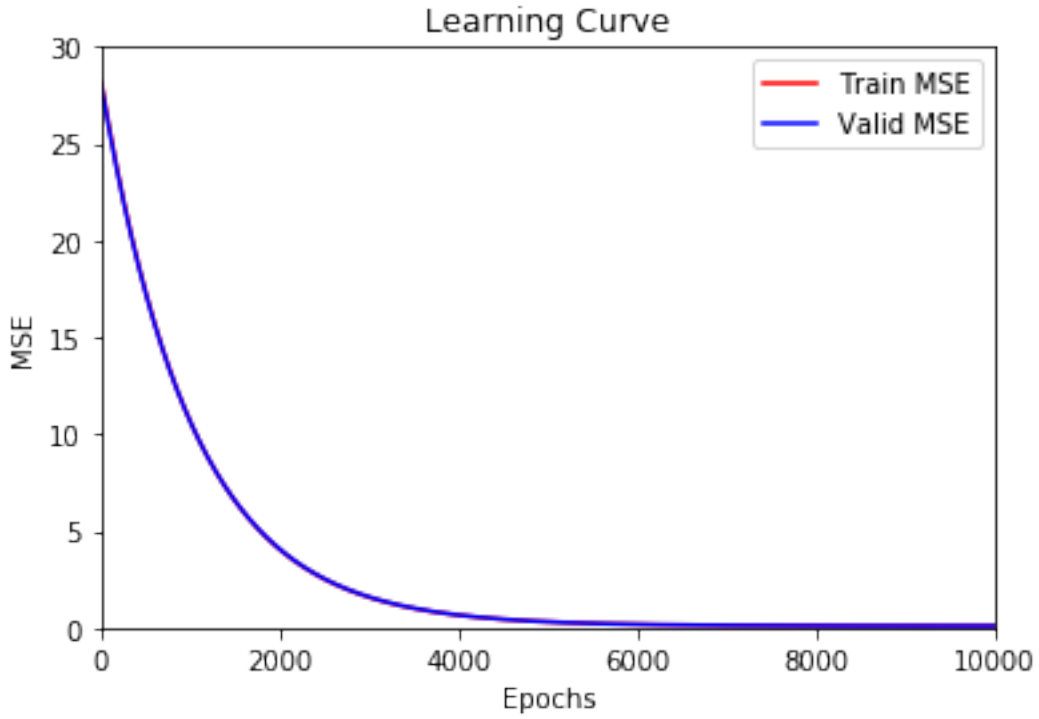
Question 3

3.1

Based on several attempts, I found that after 10000 epochs, the MSE is stabilized, so I used 10000 epochs.

```
w0 is: 4.04628545461888
w1 is: 3.7751784300755262

Training MSE: 0.10867516072717355
Validation MSE: 0.09973526899206915
At 10000 epoch
```

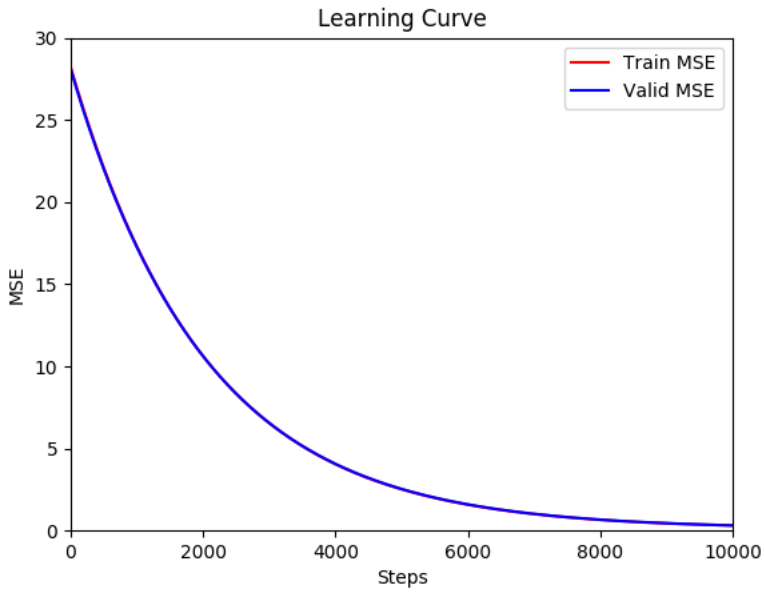


3.2

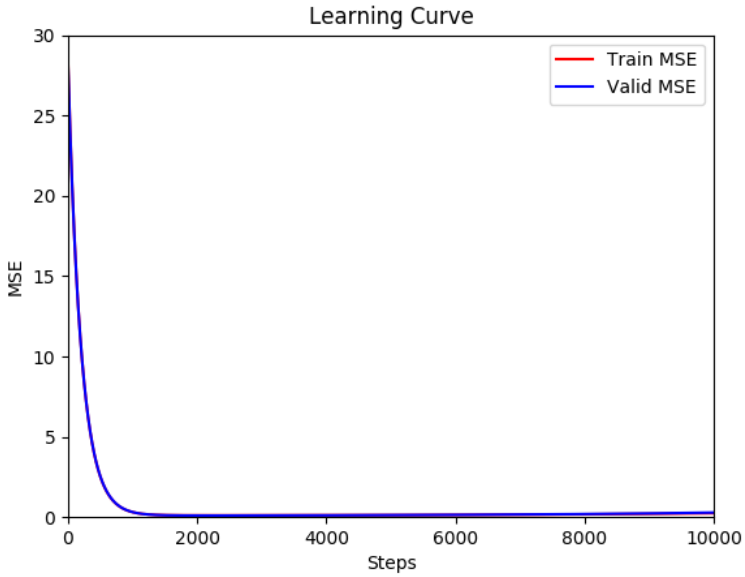
(a)

To find the best step size, we need to find which of its learning curve has the quickest drop, that is, takes minimum epochs to reach the minimum MSE.

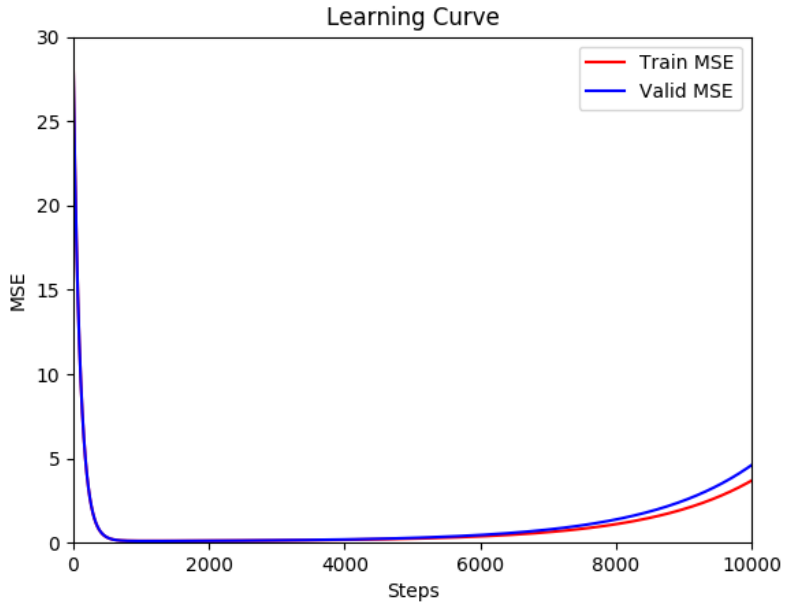
Step size = 5e-7



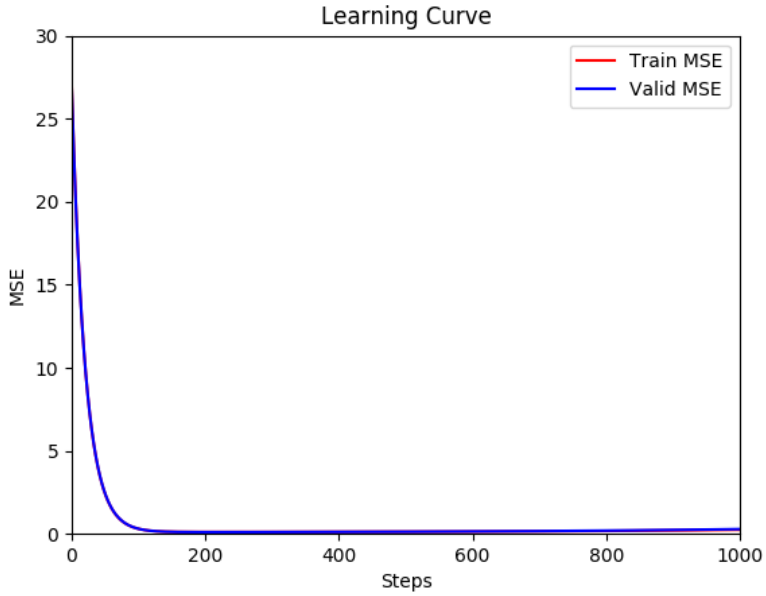
Step size = 5e-6



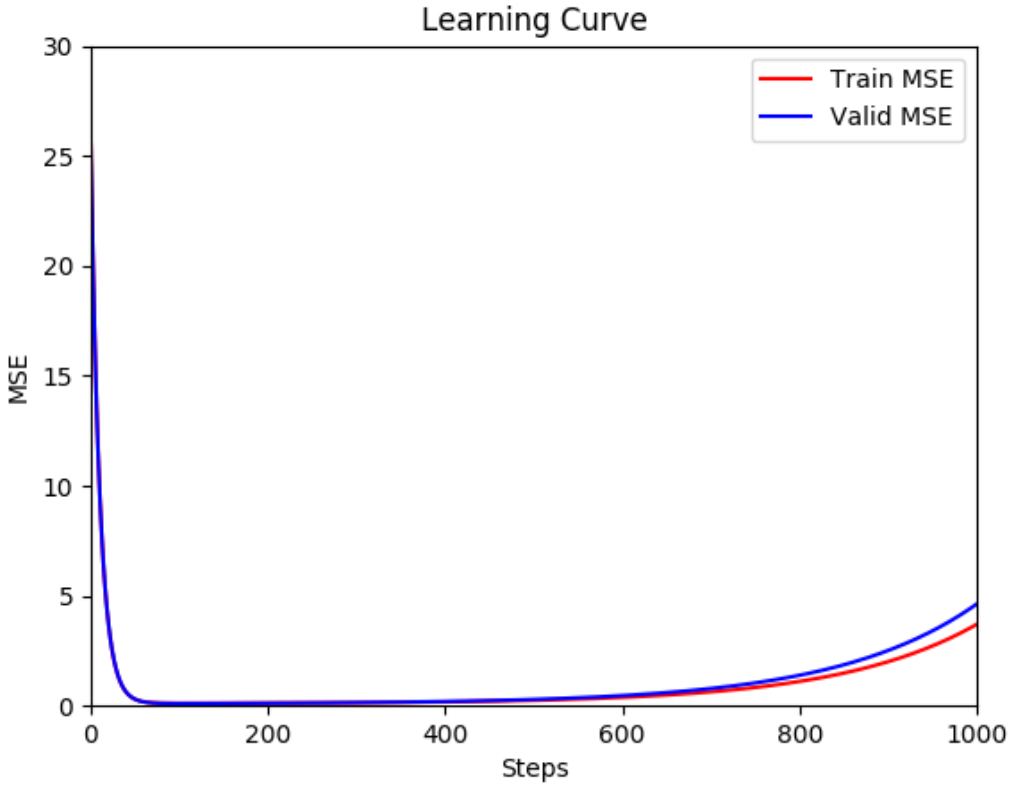
Step size = 1e-5



Step size = 5e-5



Step size = 1e-4



After several tries, we can use step size 0.0001, where the MSE drops to the minimum with minimal steps. It takes only 130 epochs to drop to MSE 0.016.

(b)

By setting step size to 1e-4 (0.0001) and epochs to 130, we get

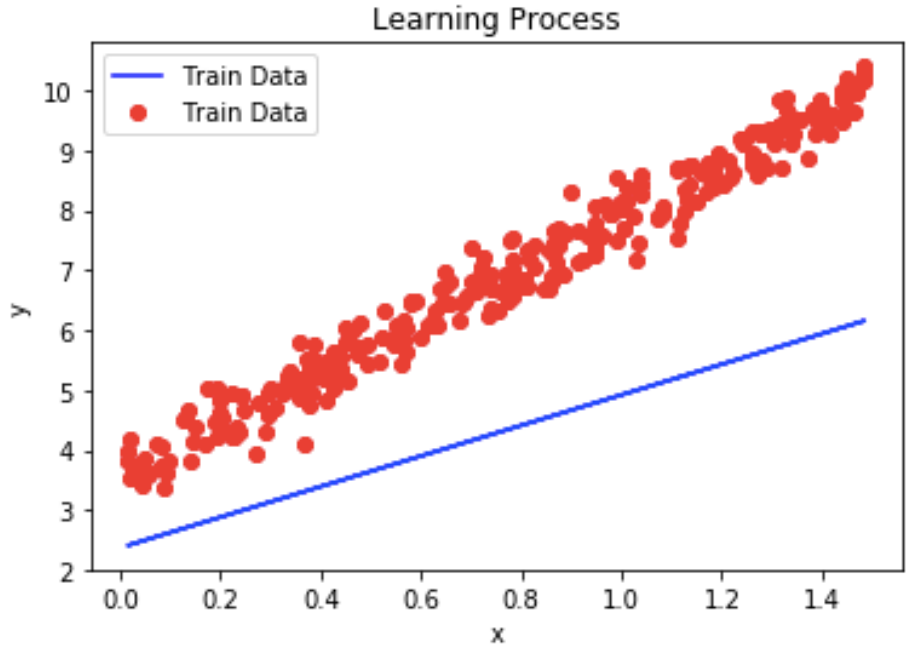
```
w0 is: 4.039179207997992
w1 is: 3.8126489977193545
with 130 epoch
0.07961865404979751
```

The MSE for test data set is around 0.08, which is really small and shows our model is good.

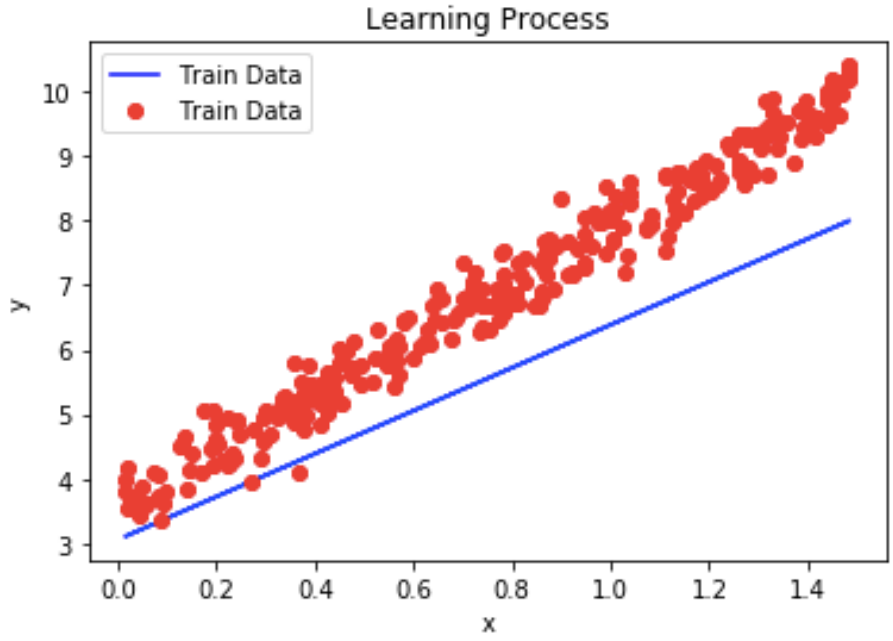
3.3

Visualization of the training process for 5 different epoch

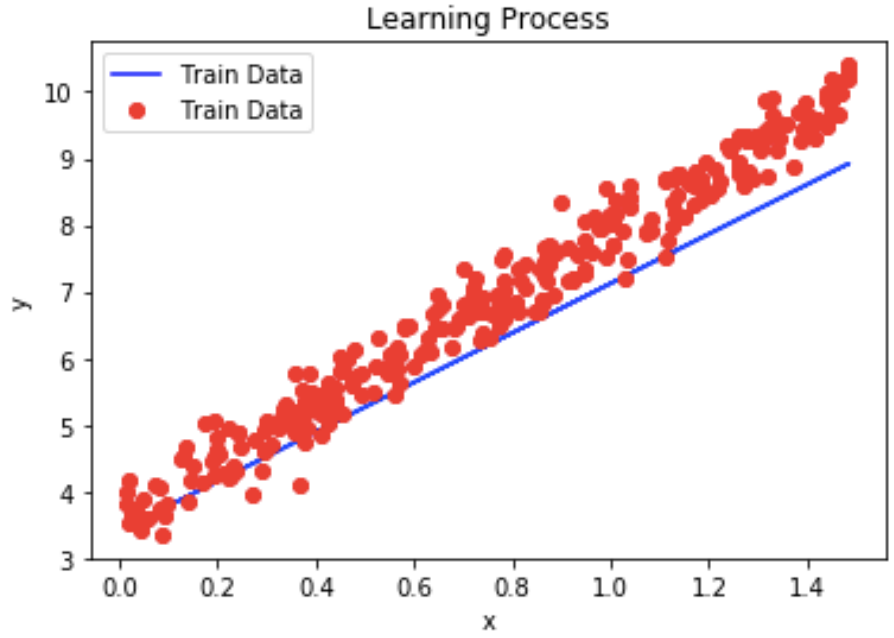
```
epoch number : 1400
```



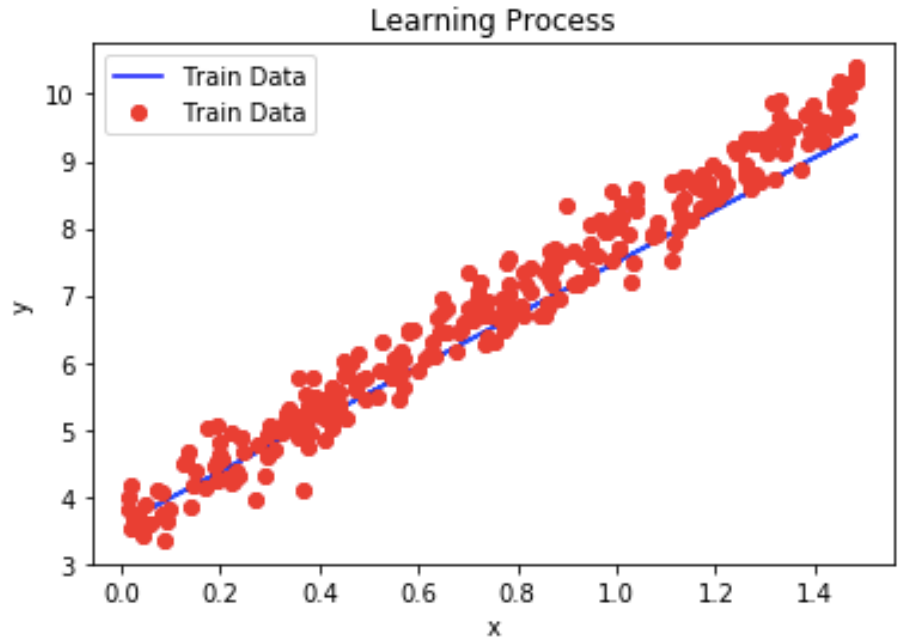

```
epoch number : 2800
```



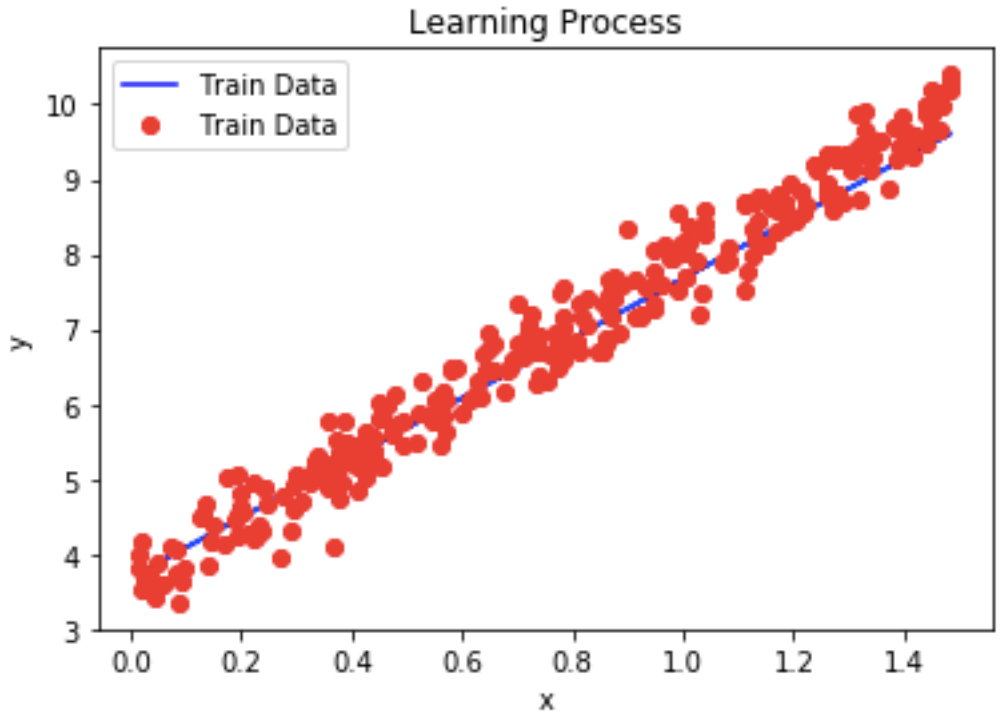
```
epoch number : 4200
```




```
epoch number : 5600
```



```
epoch number : 7000
```



We can see that as the training progresses, the fit gets better

Question 4

4.1

(a) Using sample mean is good because it does not affect the mean value of the dataset and it is easy to apply. However, mean substitution leads to bias in multivariate estimates such as regression coefficients. And if the outliers in the data set are frequent, mean substitution becomes worse.

(b) & (c) The alternatives are median imputation or regression imputation. Median imputation is good when there are great outliers. Regression imputation is a better choice, but this is based on the missing data has a small proportion. In our case we can see the missing data is 1675 out of 1994. This is a huge proportion and I believe neither of those imputations would make differences.

Thus, I will keep the mean imputation for filling the missing data.

(d) Check 'Datasets/communities.csv'

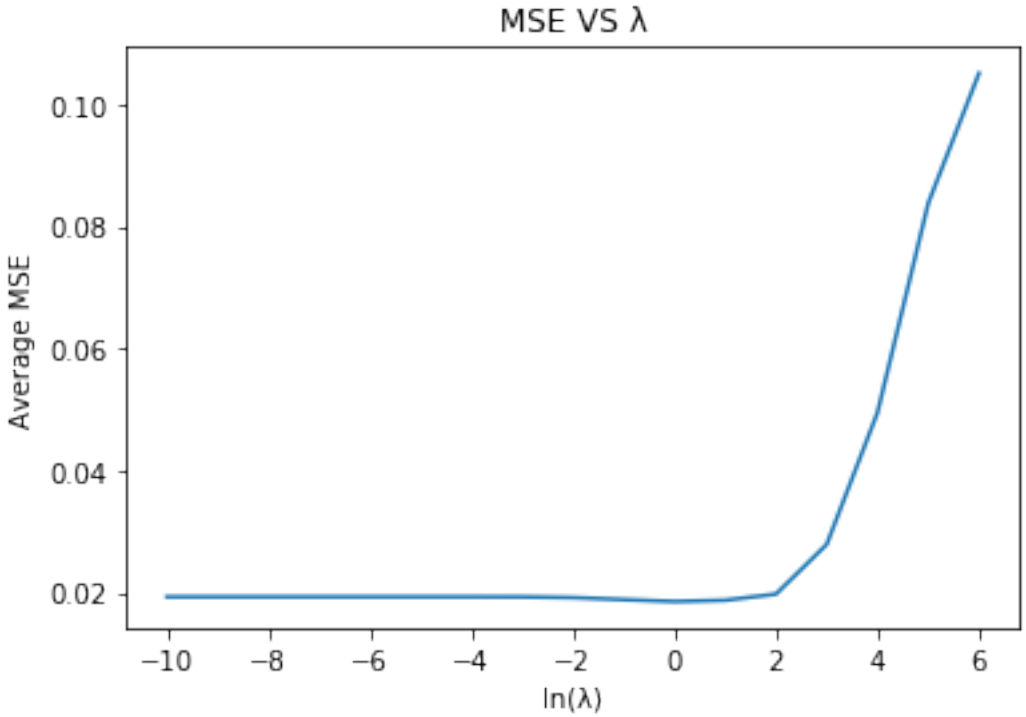
4.2

```
The MSE of number 1 80-20 split is 0.02150754016362869
The MSE of number 2 80-20 split is 0.0215948287852376
The MSE of number 3 80-20 split is 0.016454247322346405
The MSE of number 4 80-20 split is 0.01751138505010865
The MSE of number 5 80-20 split is 0.019550013245980245
The MSE averaged over 5 different 80-20 splits is 0.01932360291346032
```

Check the file 'Assignment1_260561054_4_2' for parameters learnt for each of the five splits. There are 5 columns and 123 rows (with 122 features and one bias term $x_0 = 1$).

4.2

(a)



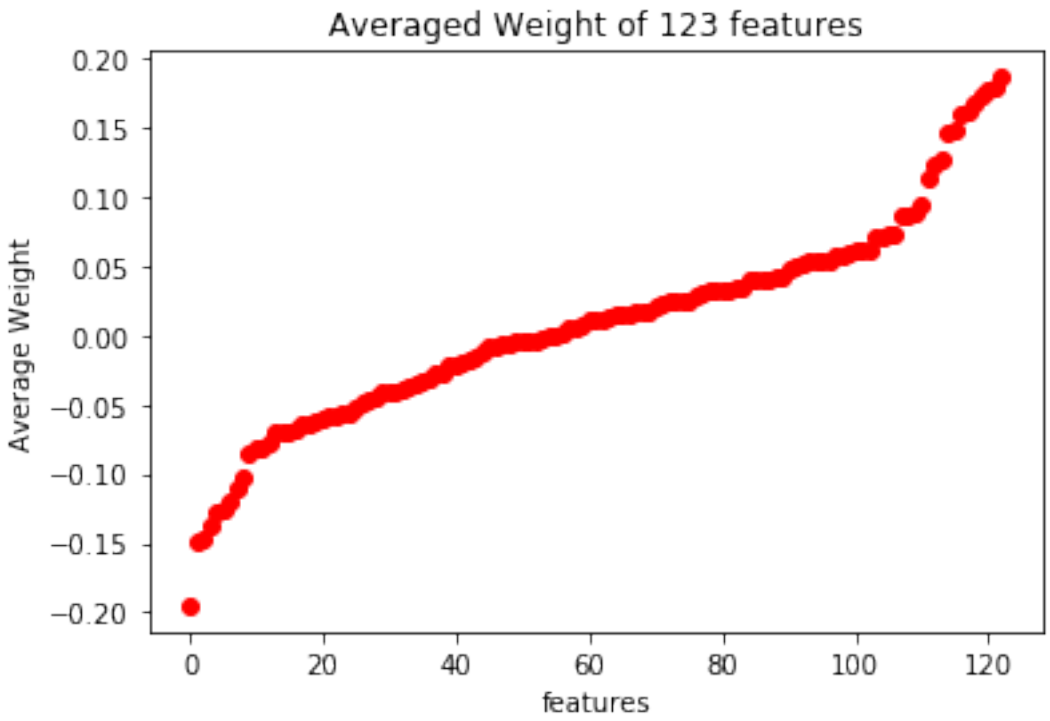

```
Minimum averaged MSE: 0.018555206975731275
λ: 1.0
```

For different values of λ from 1e-10 < λ < 1e6, the λ that gives the best fit is λ = 1.0, with average MSE = 0.018555206975731275.

Check the file 'Assignment1_260561054_4_3' for parameters learnt for each of the five splits, with λ = 1.0

(b) Yes it is possible to perform feature selection. We can perform the feature selection by removing the irrelevant features, that is, remove the features that have relative small weight.

First let's take averages for 5 different weights (parameters learnt) from 5 80-20 splits, sort the value and plot.




```
There are: 28 weights whose value is smaller than 0.02
```

From the plot, we can see that the maximum and minimum of weight are around 0.2, thus we can try remove the features whose absolute average weight values are smaller than 0.02. And there are 28 features satisfying this condition.

Then, we calculate the learning parameters ω again by cutting off those 28 irrelevant features and calculate MSEs for 5 different 80-20 splits with reduced features.

(c)

```
The MSE of number 1 80-20 split after Feature Selection is 0.0211975384711
47765
The MSE of number 2 80-20 split after Feature Selection is 0.0211576817566
50993
The MSE of number 3 80-20 split after Feature Selection is 0.0156558515055
9096
The MSE of number 4 80-20 split after Feature Selection is 0.0166315684925
80198
The MSE of number 5 80-20 split after Feature Selection is 0.0196655058866
9792
The MSE averaged over 5 different 80-20 splits after Feature Selection is
0.01886162922253357
```

(d)

Average MSE before Feature Selection: 0.018555206975731275

Average MSE after Feature Selection: 0.01886162922253357

We can see that the average MSE after Feature Selection is a little bigger than before.

The difference is around 0.0003

I think this is acceptable because the difference scale is so small thus the model performance would not be affected, which means the feature selection is successful.