Kalaiselvan Panneerselvam

278920

Lab Course Machine Learning Exercise 1

1.Exercise Sheet 1

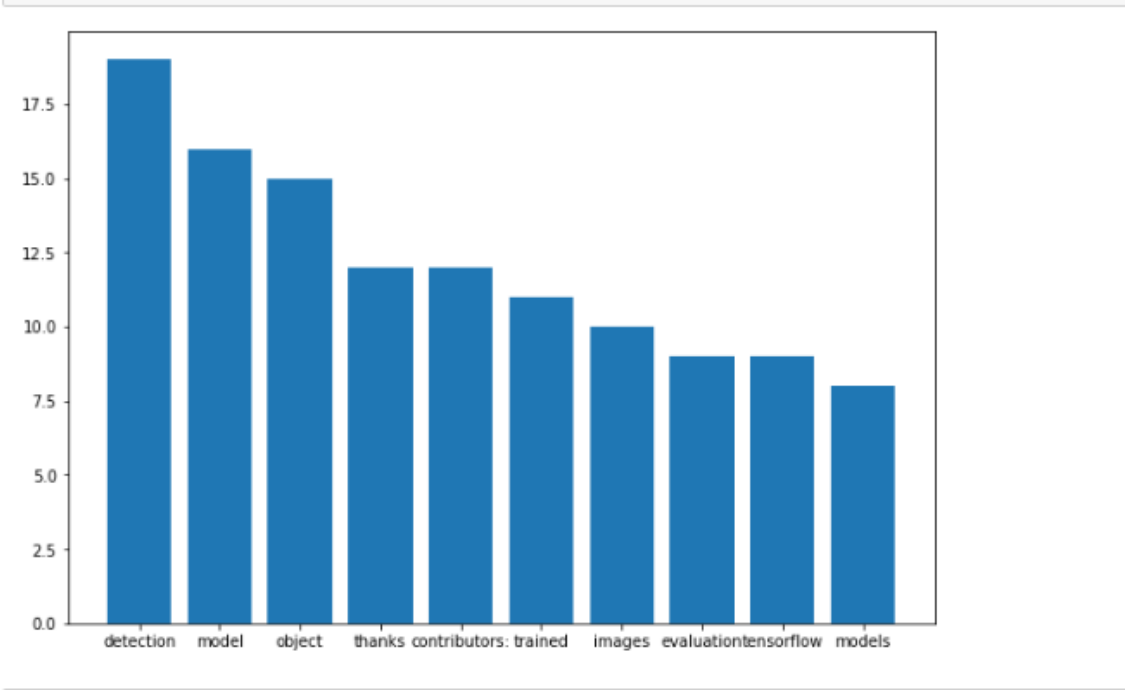
1.1 Pandas and Numpy (10 Points)

a) Word Count Program:

The list of top 10 words

```
[(19, 'detection'),
 (16, 'model'),
 (15, 'object'),
 (12, 'thanks'),
 (12, 'contributors:'),
 (11, 'trained'),
 (10, 'images'),
 (9, 'tensorflow'),
 (9, 'evaluation'),
 (8, 'models')]
```

The histogram of the top 10 words



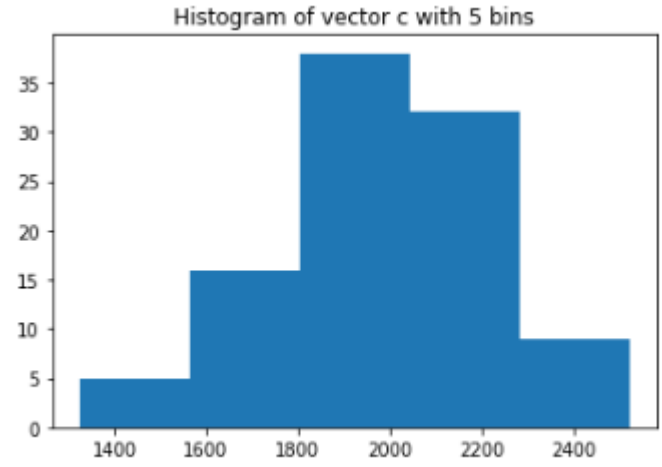
b) Matrix Multiplication:

- Iterative multiply (element-wise) each row of matrix A with vector v and sum the result of each iteration in another vector c

– Find mean and standard deviation of the new vector c

```
The new vector c is  [2263.63948123 2134.81375201 1866.15824661 1678.03251407 1974.33761921
 1804.83185072 1864.42817328 2290.55063297 2478.07487632 1602.98512219
 1668.50138487 1746.10925185 2007.56867433 1979.63880886 2054.75647506
 1899.07544071 2119.72278532 2236.63487589 1834.87917521 2084.3471475
 1646.78986894 2237.28159181 2206.06810403 1633.29657612 1573.13473236
 1326.17877225 1810.85926555 2016.59453646 2521.19670627 1829.08043346
 1825.07352729 2160.68414018 2013.26105875 1848.76541564 2251.29350536
 2259.58276933 2172.56363231 2061.72836831 2210.69479037 1883.64659169
 2185.51847054 2184.53583143 2264.75226422 1586.95801198 2280.13184166
 2517.16628636 1873.79656146 2069.79169795 1966.86170459 1924.82038447
 1955.66467125 1609.85532297 1919.29349075 1632.62365902 1952.94349316
 1638.65050947 1835.36559507 2002.86032459 2077.65092663 2123.47016594
 1956.74784302 1491.20574826 1954.32081657 2111.65441558 1480.57604705
 1373.82824623 2049.43117944 2419.46967825 1498.92524229 2005.24017459
 1940.13612619 2185.4545212  2344.7302073  2406.04702406 1611.82951296
 2272.81611259 1922.40494708 2153.37477641 1855.58938727 2281.19762094
 1829.58394381 2073.1882339  1970.37424323 2017.6388271  2203.56574736
 1883.54503081 1814.78608553 1597.95931689 1707.43000065 1651.91803858
 1973.00359442 1998.73038145 2099.40482362 1669.32974666 1941.78475115
 2313.18156887 2061.30226616 1880.11225901 2286.64537524 2217.12053704]
The mean of the vector c is 1971.851562868991
The std deviation of the vector c is 257.83185699131286
```

– Plot histogram of vector c using 5 bins



1.2 Linear Regression through exact form. (10 Points)

- Generate 3 sets of simple data. i.e. a matrix A with dimensions 100 × 2. Initialize it with normal distribution μ = 2 and σ = [0.01, 0.1, 1].

```
one=np.random.normal(2, 0.01, (100, 2))
two=np.random.normal(2, 0.1, (100, 2))
three=np.random.normal(2, 1, (100, 2))
```

- Implement LEARN-SIMPLE-LINREG algorithm and train it using matrix A to learn values of β0 and β1.

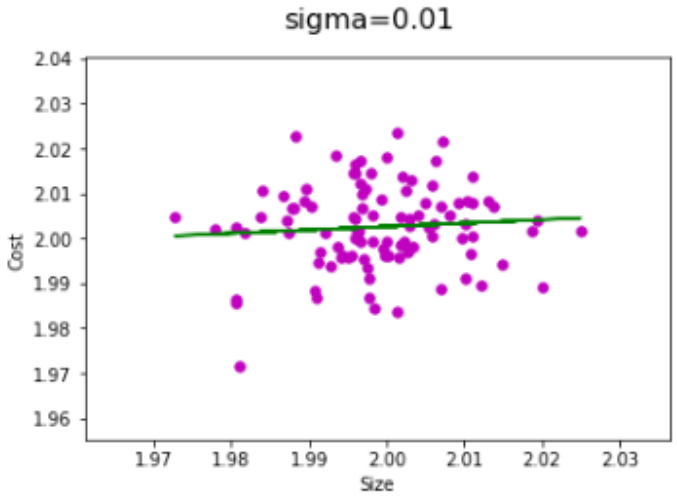
```
def LEARNSIMPLELINREG(x,y):
    x_mean, y_mean = mean(x), mean(y)
    b1 = covariance(x, x_mean, y, y_mean) / variance(x, x_mean)
    b0 = y_mean - b1 * x_mean
    return [b0, b1]
```

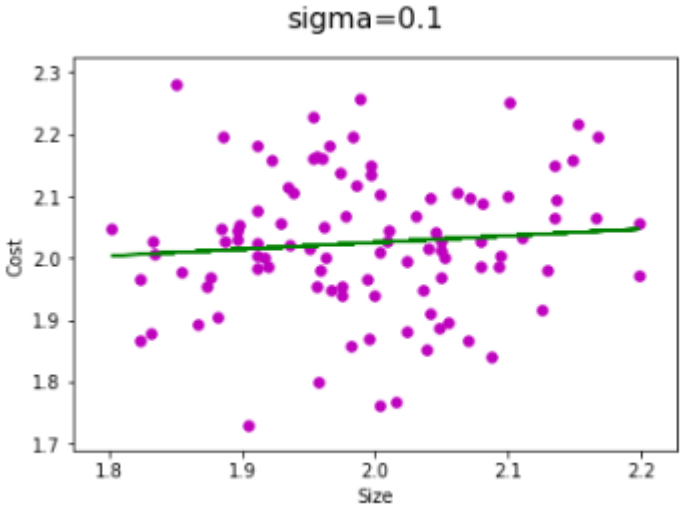
- Implement PREDICT-SIMPLE-LINREG and calculate the points for each training example in matrix A.

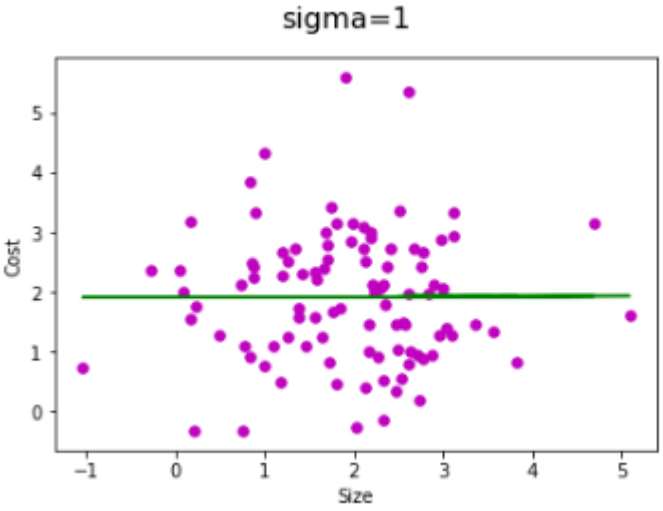
```
def PREDICTSIMPLELINREG(x, y, b):
    # predicted response vector
    y_pred = b[0] + b[1]*x
    return y_pred
```

-Plot the training points from matrix A and predicted values in the form of line graph.

```
The intercept and slope when sigma=0.01 is [1.8514653567333181, 0.07557042708625683]
The intercept and slope when sigma=0.1 is [1.8076333604445547, 0.10866941362639547]
The intercept and slope when sigma=1 is [1.9111609027440013, 0.004048891542928851]
The RMSE when sigma=0.01 is 7.783495247125329e-05
The RMSE when sigma=0.1 is 0.00071025305956876
The RMSE when sigma=1 is 0.00678520116701255
```







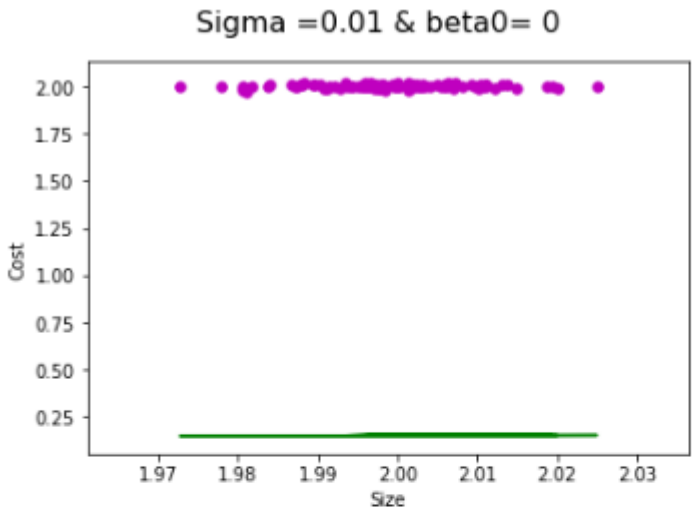
-Comment on the effect that σ has on the line that is predicted.

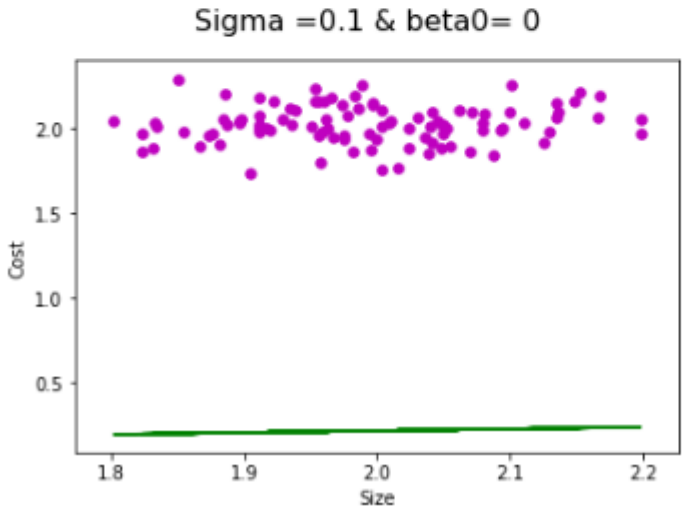
```
The intercept and slope when sigma=1 is [1.9111609027
The RMSE when sigma=0.01 is 7.783495247125329e-05
The RMSE when sigma=0.1 is 0.00071025305956876
The RMSE when sigma=1 is 0.00678520116701255
```

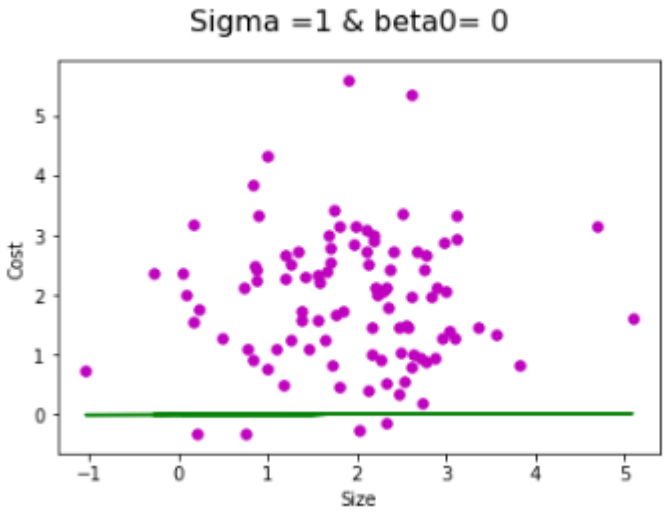
When sigma value increases from 0.01 to 1, the RMSE also increases. So its always good to have lower sigma value like 0.01.

-Put β0 to zero and rerun the program to generate the predicted line. Comment on the change you see for the varying values of σ.

When β0 is equal to zero , the linear equation becomes y(x)= β1(x). The prediction becomes directly proportional to the X value. When values of sigma is increased it can be seen that RMSE also increases. The model becomes fully dependent on the x values which in turn gives better predictions.

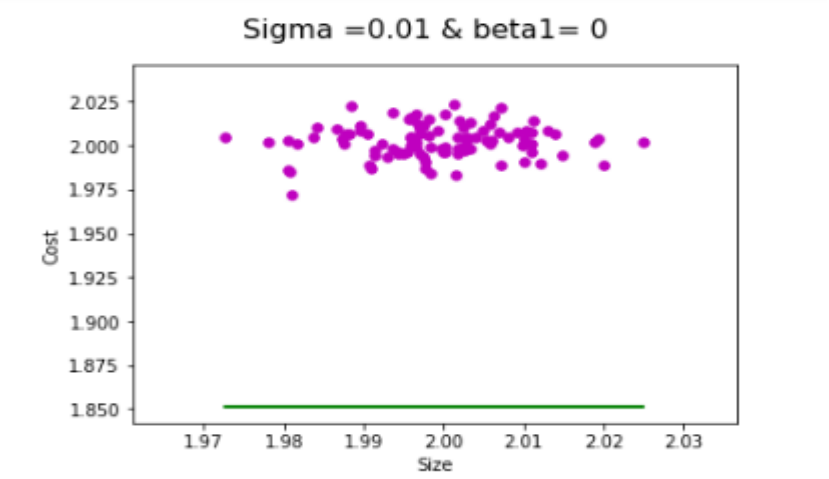


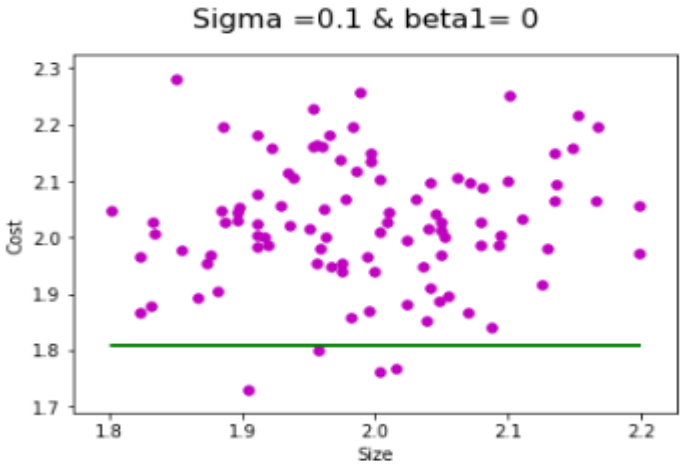


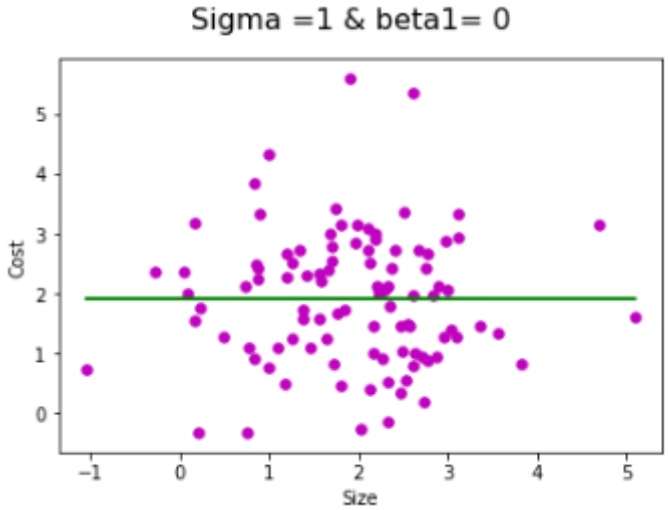


- Put β1 to zero and rerun the program to generate the predicted line. Comment on the change you see for the varying values of σ.

When β1 is equal to zero , the linear equation becomes y(x)= β0. The prediction becomes directly proportional to the constant value. The model becomes fully dependent on the constant values thereby gives poor predictions.







- In the end use numpy.linalg lstsq to replace step 2 for learning values of β0 and β1.

```
m1,c1 = np.linalg.lstsq(A1, y1)[0]
m2,c2 = np.linalg.lstsq(A2, y2)[0]
m3,c3 = np.linalg.lstsq(A3, y3)[0]
print("The intercept and slope when sigma=0.01 is", c1,m1)
print("The intercept and slope when sigma=0.1 is", c2,m2)
print("The intercept and slope when sigma=1 is", c3,m3)
```

```
The intercept and slope when sigma=0.01 is 1.8514653567333041 0.07557042708626495
The intercept and slope when sigma=0.1 is 1.8076333604445594 0.10866941362639325
The intercept and slope when sigma=1 is 1.9111609027440029 0.0040488915429286905
```

Using the built in function numpy.linalg lstsq calculated the estimator values beta0 and beta1 which is same as that of the values returned by LEARN_SIMPLE_LINREG function.