1. Give one example of how machine learning can be used in the following industries or applications? This question may be done as a group. You may consult the Internet. Please be ready to share your answers at the tutorial session.
    - Retail
    - Banking
    - Healthcare
    - Fashion
    - Education
    - Communication Networks
    - Smart Home
    - Social Media

    Answer
    Retail: to predict the demand for a product based on sales data.
    Banking: to assess personal creditworthiness on a loan.
    Healthcare: to evaluate and predict the treatment effect of medicine based on experimental data.
    Fashion: to predict the trends of a cosmetic style based on sales data.
    Education: to assess the effect of a teaching strategy based on academic data.
    Communication Network: to identify the spam contents based on those marked as spam data.
    Smart Home: to predict the activities of the host based on their daily activity records.
    Social Media: to push customized content based on the viewed content of a user.

2. In the lectures, we have discussed several examples of supervised learning, unsupervised learning and reinforcement learning problems. Give two examples of each type of problem that is different from the examples discussed in the lectures. This question may be done as a group. You may consult the Internet. Please be ready to share your answers at the tutorial session.

    Answer
    Supervised learning:
    1) Housing price prediction based on decision trees.
    2) Patient classification based on Naïve Bayes.

    Unsupervised learning:
    1) Classify customers without providing customer targets.
    2) DNA/gene classification.

    Reinforcement learning:
    1) AI-based chess game
    2) Self-driving vehicle

3. Consider the tire tread versus mileage problem we discussed in the lecture. The problem is to predict the tire tread depth from the mileage. The dataset, which has nine pairs of points, is reproduced below. This is an individual assignment to be done by every student. You may work as a group or get help from others, but I expect every student to implement the code themselves.

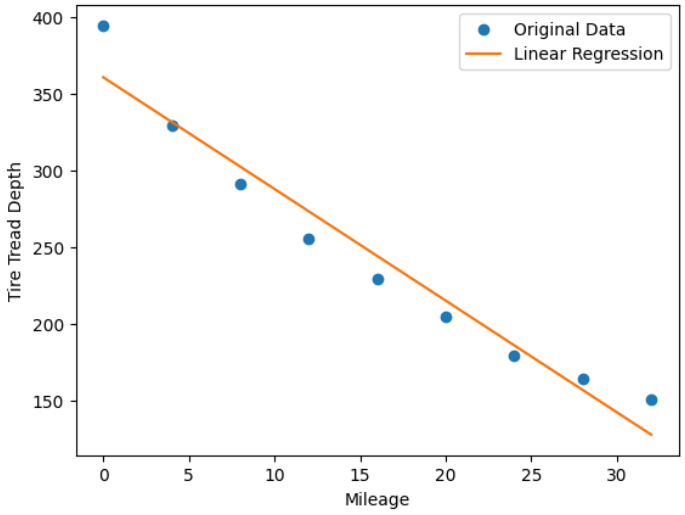
| Mileage (in 1000 miles) $x$ | Tire Tread Depth (in mils) $y$ |
|---|---|
| 0 | 394.33 |
| 4 | 329.50 |
| 8 | 291.00 |
| 12 | 255.17 |
| 16 | 229.33 |
| 20 | 204.83 |
| 24 | 179.00 |
| 28 | 163.83 |
| 32 | 150.33 |

a. Compute the linear regression solution (i.e., best fit line) for this dataset. Use the entire dataset to train and find the best fit line. Give the expression for the best fit line and compute the error performance on the training dataset. Recall that the error performance is measured by the sum of squared errors. For this question, you can use Python to do the computations, but you may not use Scikit-learn to do the regression.
b. Plot the best fit line over the data points and comment on whether the fit is good.
c. Leave out the last sample (x=32) and use it as a test data point. Use the remaining samples to train and find the best fit line. Give the expression for the best fit line. Compute the training error and the test error performance.
d. Using the entire dataset to train, find the linear regression solution using Scikit-learn and compare to the solution you got in part (a). The two solutions might be different. Can you guess why?

Answers

a. y = -7.28x + 360.64
squared error = [1135.02, 4.08, 129.96, 327.97, 219.93, 104.24, 47.89, 49.42, 513.02]
sum of squared error = 2531.53 mean squared error = 281.28

b. The regression plot is shown in the figure below.



The original data contain tuples that potentially need to be deprecated (outliers), for instance, (0, 394.33) and (32, 150.33). Therefore, this regression might not be accurate.

c. y = -7.89x + 366.3
sum of squared error of training = 1705.33 mean squared error of training = 213.17
sum of squared error of testing = 3038.31 mean squared error of testing = 337.59

d. y = -7.28x + 360.64, which is the same as the one in task a.

Theoretically, there should be better performance of Sklearn's linear regression because it normalizes the data before implementing the basic least square method. The implementations of both two algorithms are demonstrated as follows.

The linear regression algorithm used in tasks a-c is a one-dimensional linear regression that only applies the basic least square method, which determines the mathematical expression of the target regression line, namely,

$$y = \hat{b}x + \hat{a}, \text{ where } \hat{b} = \sum \frac{(x_i - \overline{x})(y_i - \overline{y})}{(x_i - \overline{x})^2}, \text{ and } \hat{a} = \overline{y} - \hat{b}\overline{x}$$

Based on the equations above, the least square method in Python can be implemented by the Python code below.

```
def linear_regression(x, y):
    length = len(x)
    x_sum = 0
    for i in range(0, length):
        x_sum += x[i]
    x_avg = x_sum / length

    y_sum = 0
    for i in range(0, length):
        y_sum += y[i]
    y_avg = y_sum / length

    p1 = 0
    p2 = 0
    for i in range(0, length):
        p1 += (x[i] - x_avg)*(y[i] - y_avg)
        p2 += (x[i] - x_avg)*(x[i] - x_avg)

    b = p1 / p2
    a = y_avg - b*x_avg

    a = round(a, 2)
    b = round(b, 2)

    return([a, b])
```

There are 3 types of scipy linear regression algorithms used in Sklearn's linear regression pack, among which corresponds to this case is optimized.nnls() method, which is a non-

negative least square method with a data pre-processing phase. This phase can be checked in Scipy's optimize pack. It can be easily found that there is a normalization in nnls method.

```
x, rnorm, mode = __nnls.nnls(A, m, n, b, w, zz, index, maxiter)
if mode != 1:
    raise RuntimeError("too many iterations")

return x, rnorm
```

Generally, normalization is used to scale data and limit all data to the range [0, 1]. The advantage of using normalization before regression is that it can weaken the influence of labeled data when the gap between the maximum and minimum values in the data is much larger than the step size of the dataset, especially if outliers are existing in the dataset.

However, given the small data range and volume of this dataset, the benefits of normalization are not well performed in this case.