P1.a. Design a simple machine learning model to train the training i and test the same.

P1.a. Design a simple machine learning model to train the training instances and test the same.

```
# Generating the Training Set

# python library to generate random numbers
from random import randint

# the limit within which random numbers are generated
TRAIN_SET_LIMIT = 1000

# to create exactly 100 data items
TRAIN_SET_COUNT = 100

# list that contains input and corresponding output
TRAIN_INPUT = list()
TRAIN_OUTPUT = list()
```

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

```
# loop to create 100 data items with three columns each
for i in range(TRAIN_SET_COUNT):
    a = randint(0, TRAIN_SET_LIMIT)
    b = randint(0, TRAIN_SET_LIMIT)
    c = randint(0, TRAIN_SET_LIMIT)

# creating the output for each data item
    op = a + (2 * b) + (3 * c)
    TRAIN_INPUT.append([a, b, c])

# adding each output to output list
    TRAIN_OUTPUT.append(op)
```

```
# printing first 10 records
TRAIN_OUTPUT[:10]
```

```
[3741, 2785, 4423, 4865, 4154, 1860, 1653, 4168, 3176, 1541]
```

```
# printing first 10 records
TRAIN_INPUT[:10]
```

```
[[259, 337, 936],
 [376, 984, 147],
 [958, 690, 695],
 [935, 657, 872],
 [806, 387, 858],
 [819, 63, 305],
 [895, 295, 56],
 [120, 635, 926],
 [223, 332, 763],
 [512, 72, 295]]
```

```
# Training the Model
# The data that was created using the above code is used to train the model

# Sk-Learn contains the linear regression model
from sklearn.linear_model import LinearRegression

# Initialize the linear regression model
predictor = LinearRegression(n_jobs =-1)

# Fill the Model with the Data
predictor.fit(X = TRAIN_INPUT, y = TRAIN_OUTPUT)
```

```
LinearRegression
LinearRegression(n_jobs=-1)
```

Start coding or generate with AI.

The Model can be created in two steps:-

Training the model with Training Data

Testing the model with Test Data

LinearRegression(n_jobs=-1) Testing the Data

The testing is done Manually. Testing can be done using some random data and testing if the model gives the correct result for the input data.

```
# Random Test data
X_TEST = [[ 10, 20, 30 ]]   #---> 10 + 20*2 + 30*3 = 140.

# Predict the result of X_TEST which holds testing data
outcome = predictor.predict(X = X_TEST)

# Predict the coefficients
coefficients = predictor.coef_

# Print the result obtained for the test data
print('Outcome : {}\nCoefficients : {}'.format(outcome, coefficients))
```

```
Outcome : [140.]
Coefficients : [1. 2. 3.]
```