**Regression**

Box 1

X — House size from 1K sq feet to 10K sq feet.

Y — Cost of the house from 300K to 1200K.

Box 2

1. *numpy.array()* short form as np.array()

2.matplotlib.pyplot.plot(x,y) short form as plt.plot(x,y),these are for ease of coding.

Box 3

The aim is to find the **best fit line**, which **minimizes error** by finding the sum of the square of the distance between points and the line

Box 4

Gradient descent is the best here to find the least error when compared to other instances.

**Classification**

Box 1

Scikit-learn is a python library that can be used for machine learning models.

Box 2

Binary classification is where we wish to group an outcome into one of two groups.

Box 3

To fit a binary logistic regression with sklearn, we use the [LogisticRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.htmlLogisticRegression" \t "_blank) module with multi\_class

Box 4

Libraries such as Tensorflow and Keras are also used for fitting and customizing Neural Networks

**Clustering**

Box 1

KNN is [lazy learning](https://en.wikipedia.org/wiki/Lazy_learning) algorithm and therefore requires no training prior to making real time predictions.

Box 2

There are only two parameters required to implement KNN i.e. the value of K and the distance function

Box 3

The above script splits the dataset into 80% train data and 20% test data.

Box 4

The gradient descent algorithm also converges faster with normalized features.

Box 5

To train the KNN import the KNeighborsClassifier class from the sklearn.neighbors library.

Box 6

In each iteration the mean error for predicted values of test set is calculated and the result is appended to the error list.