Machine Learning

* [Data Science and Machine Learning with Python – Hands On!](https://click.linksynergy.com/link?id=ENhIHOsZQ7Y&offerid=323058.671576&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Fdata-science-and-machine-learning-with-python-hands-on%2F)https://ad.linksynergy.com/fs-bin/show?id=ENhIHOsZQ7Y&bids=323058.671576&type=2&subid=0
* [Introduction to Machine Learning & Face Detection in Python](https://click.linksynergy.com/link?id=ENhIHOsZQ7Y&offerid=323058.617930&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Fintroduction-to-machine-learning-in-python%2F)https://ad.linksynergy.com/fs-bin/show?id=ENhIHOsZQ7Y&bids=323058.617930&type=2&subid=0

Machine Learning:

* [Linear Regression](https://pythonspot.com/linear-regression/)
* [k nearest neighbors](https://pythonspot.com/k-nearest-neighbors/)
* [Support Vector Machine](https://pythonspot.com/en/support-vector-machine/)
* [Supervised Learning](https://pythonspot.com/en/supervised-learning/)
* [Deep Learning](https://pythonspot.com/en/deep-learning/)

**Machine Learning Introduction**  
Machine Learning is essentially to make predictions or behaviors based on data. Given a Machine Learning System , it will do a certain behavior or make predictions based on data.  
  
The data for a Machine Learning System entirely depends on the problem to be solved. Different problems, different datasets.  
  
The desired output usually falls into one of these categories:

* prediction
* clustering
* classification
* regression
* recommendation

Some practical examples of problems that may be approached with Machine Learning algorithms are:

* optical character recognition (OCR)
* computer vision
* search engines
* speech recognition
* recommender systems
* financial market analysis

# Supervised Learning

**Supervised Learning Phases**  
All supervised learning algorithms have a training phase (supervised means ‘to guide’). The algorithm uses training data which is used for future predictions.

# 

# 

The supervised learning process always has 3 steps:

* build model (machine learning algorithm)
* train mode (training data used in this phase)
* test model (hypothesis)

**Examples**  
In Machine Learning, an example of supervised learning task is **classification**. Does an input image belong to class A or class B?  
  
A specific example is ‘face detection’. The training set consists of images containing ‘a face’ and ‘anything else’. Based on this training set a computer may detect a face (more similar to features from one set compared to the other set).  
  
Application of supervised learning algorithms include:

* Financial applications (algorithmic trading)
* Bioscience (detection)
* Pattern recognition (vision and speech)

# Linear Regression

How does **regression** relate to **machine learning**?

Given data, we can try to find the best fit line. After we discover the best fit line, we can use it to make predictions.

Consider we have data about houses: price, size, driveway and so on. You can [download the dataset for this article here](https://vincentarelbundock.github.io/Rdatasets/csv/Ecdat/Housing.csv).

Data can be any data saved from Excel into a csv format, we will use Python Pandas to load the data.

**Required modules**  
You shoud have a few modules installed:

|  |
| --- |
| sudo pip install sklearn  sudo pip install scipy  sudo pip install scikit-learn |

**Load dataset and plot**  
You can choose the graphical toolkit, this line is optional:

|  |
| --- |
| matplotlib.use('GTKAgg') |

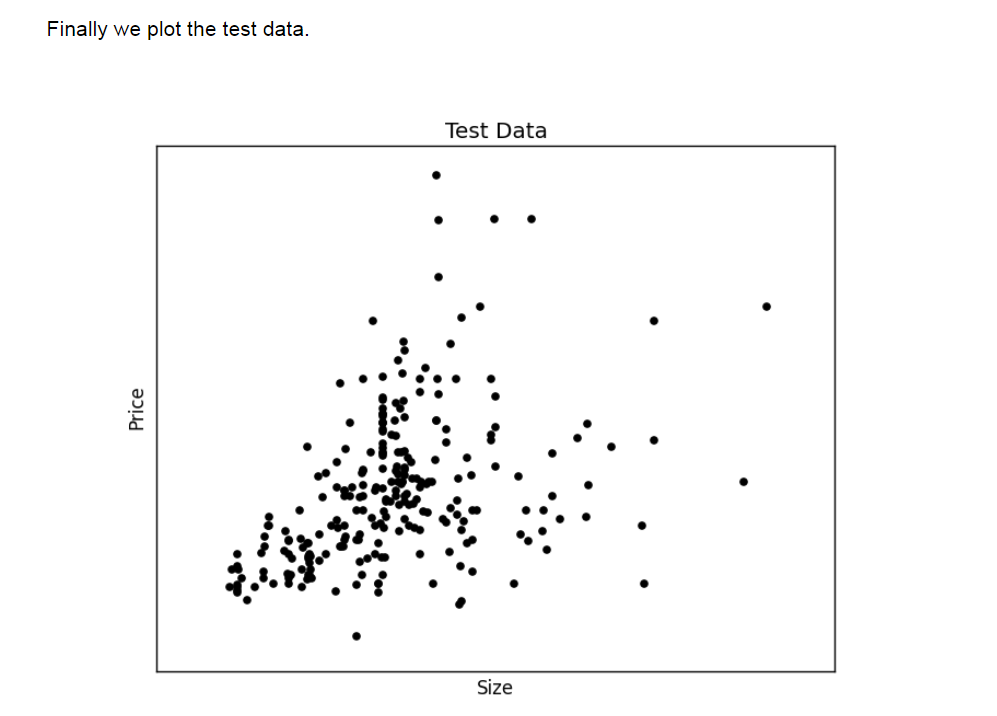
We start by loading the modules, and the dataset. Without data we can’t make good predictions.

The first step is to load the dataset. The data will be loaded using Python Pandas, a data analysis module. It will be loaded into a structure known as a Panda Data Frame, which allows for each manipulation of the rows and columns.

We create two arrays: X (size) and Y (price). Intuitively we’d expect to find some correlation between price and size.

The data will be split into a trainining and test set. Once we have the test data, we can find a **best fit line** and **make predictions**.

|  |
| --- |
| **import** matplotlib  matplotlib.use('GTKAgg')    **import** matplotlib.pyplot **as** plt  **import** numpy **as** np  **from** sklearn **import** datasets, linear\_model  **import** pandas **as** pd    *# Load CSV and columns*  df = pd.read\_csv("Housing.csv")    Y = df['price']  X = df['lotsize']    X=X.reshape(len(X),1)  Y=Y.reshape(len(Y),1)    *# Split the data into training/testing sets*  X\_train = X[:-250]  X\_test = X[-250:]    *# Split the targets into training/testing sets*  Y\_train = Y[:-250]  Y\_test = Y[-250:]    *# Plot outputs*  plt.scatter(X\_test, Y\_test, color='black')  plt.title('Test Data')  plt.xlabel('Size')  plt.ylabel('Price')  plt.xticks(())  plt.yticks(())    plt.show() |



We have created the two datasets and have the test data on the screen. We can continue to create **the best fit line**:

|  |
| --- |
| *# Create linear regression object*  regr = linear\_model.LinearRegression()    *# Train the model using the training sets*  regr.fit(X\_train, Y\_train)    *# Plot outputs*  plt.plot(X\_test, regr.predict(X\_test), color='red',linewidth=3) |



To make an individual prediction using the linear regression model:

|  |
| --- |
| **print**( str(round(regr.predict(5000))) ) |