# Supervised Learning

## Introduction

> *In supervised machine learning the data comes with an attribute that we would like to predict called the* label, target or dependent variable *usually noted with y, this attribute is labelled in supervised learning and unlabelled in unsupervised learning.*
>
> *It also comes with a* feature or features *which are known as the* independent variable *a feature cannot affect another feature noted as X for a set or x for a single element.*

## Types

### Classification

> *The data samples belong to different categories in other words, your features are*

### Regression

> *The data is continuous and*

## Steps

### 1. Import the necessary libraries

### 2. Load the data

COMMON READ METHODS

```
import pandas as pd

pd.read_csv('filepath', sep = ',', delimeter = 'None', delim_whitespace = Fal
pd.read_table('filepath', sep = '\t', header = None)
pd.read_sql('filepath')
pd.read_excel('filepath')
```

### 3. Manipulate and explore the data

> *data splitting for a DataFrame*

```
X = data[0,1]
y = data[2]
```

### 4. Split the data

> *You should train the model on a certain data set then measure its accuracy by using a completely different set. This is achieved by splitting your data into two sets the* `training set` *and the* `testing set`*. but before that you should split your loaded data into features* `X` *and a label* `y` *.*
> *The code below assumes that you have your X and y variables ready.*

IMPORT STATEMENTS AND USAGE

```
from sklearn.model_selection import train_test_split

# note that test_size and random_state are optional parameters.
# the test_size indicates the percentage of the testing set.
# in the code below our training set will be 75% of the entire data set.
# the parameter random_state will split the data randomly if not specified.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, r
```

### 5. Choose the model

### 6. Fit the model

> *This is where you actually train your model, you will fit your training sets*

CODE EXAMPLE

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()

model.fit(X_train, y_train)
```

# Linear Regression

# Perceptron Algorithm

## Introduction

# Decision Trees

Decision Trees

> *Decision Trees can be thought of as asking yes no questions and seeing the path or the branching of the tree*
>
> *85*

# Naïve Bayes

# Support Vector Mechanics

# Ensemble Methods

# Model Evaluation Metrics

# Training and Tuning

## Types of Error

## KFold Cross Validation

```
from sklearn.model_selection import KFold

X = range(12)
kf = KFold(3, shuffle = true)

for train, test in kf.split(X):
    print(train, test)
```