Machine Learning

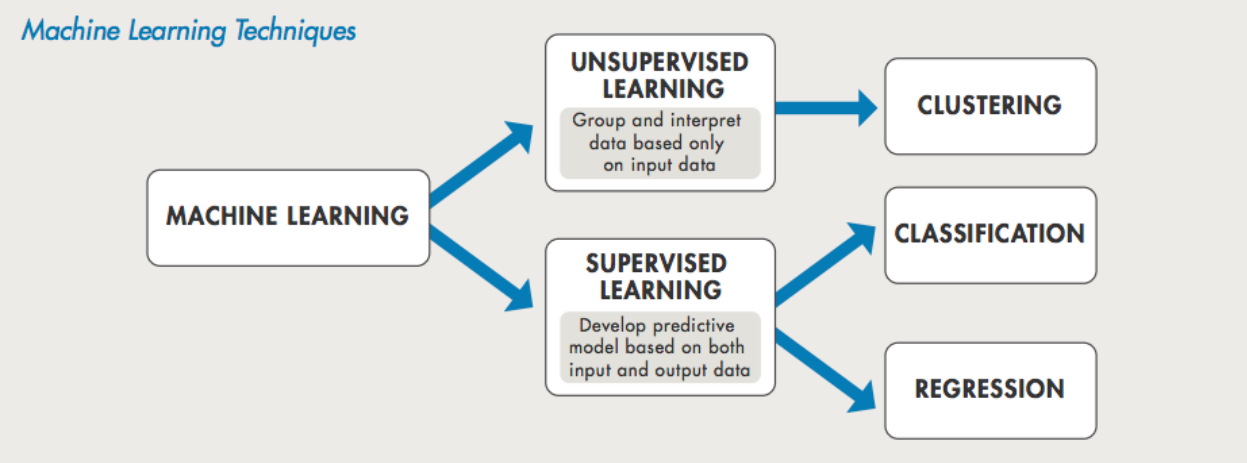
week 1 :

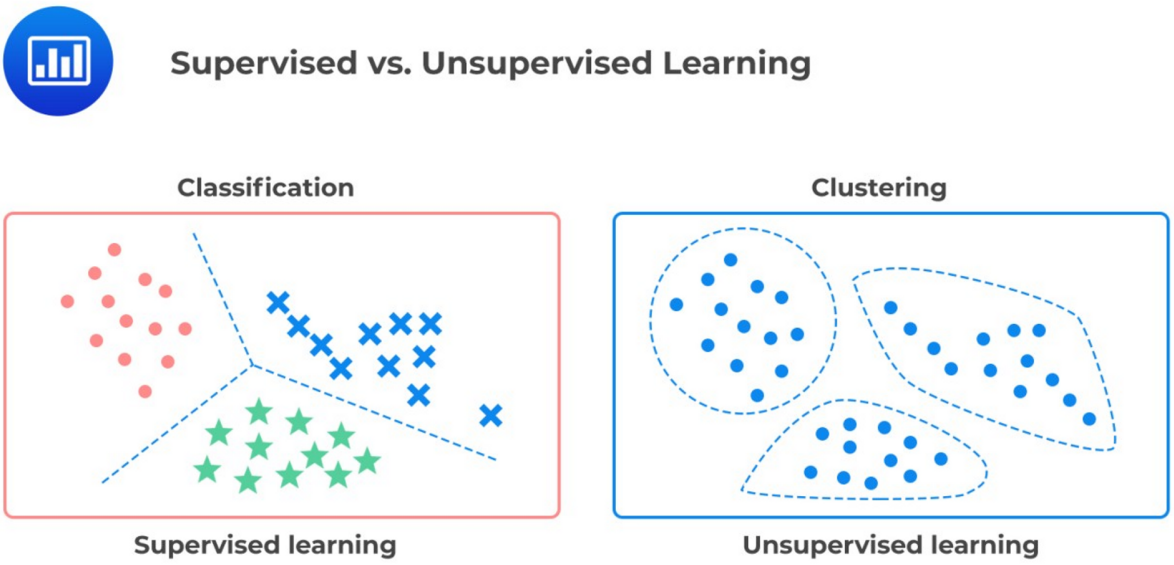
MACHINE LEARNING

(Informal)

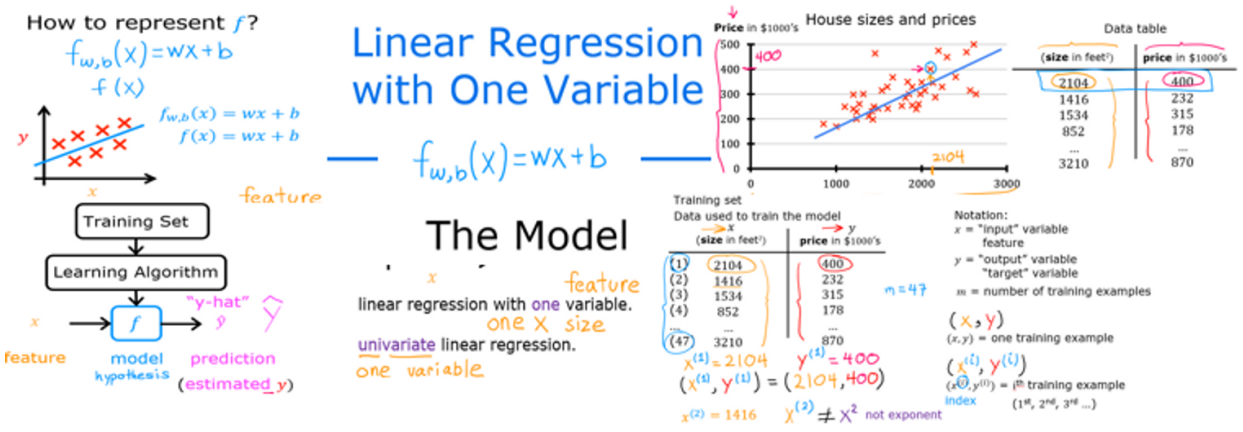
A subfield of AI that gives computers the ability to

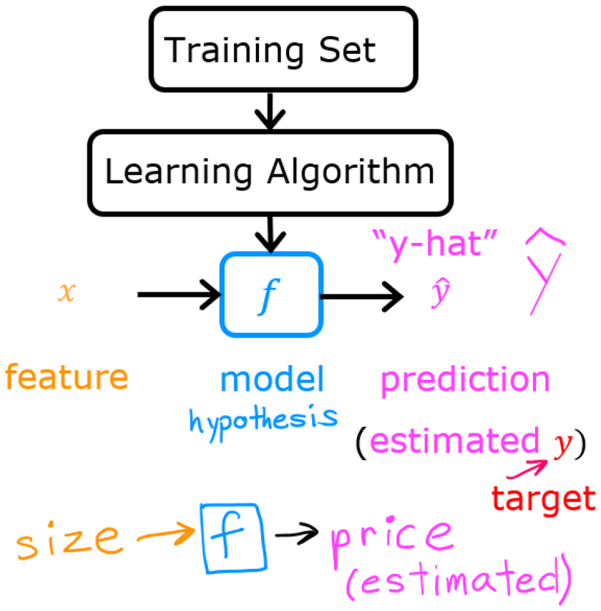
learn without being **explicitly programmed.**


Machine Learning Techniques
MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpret data based only on input data
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLUSTERING
CLASSIFICATION
REGRESSION


Supervised vs. Unsupervised Learning
Classification
Supervised learning
Clustering
Unsupervised learning

Linear Regression

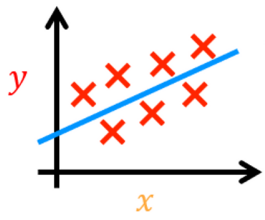




How to represent f?

$f_{w,b}(x) = wx + b$

$f(x)$



$f_{w,b}(x) = wx + b$

$f(x) = wx + b$

feature

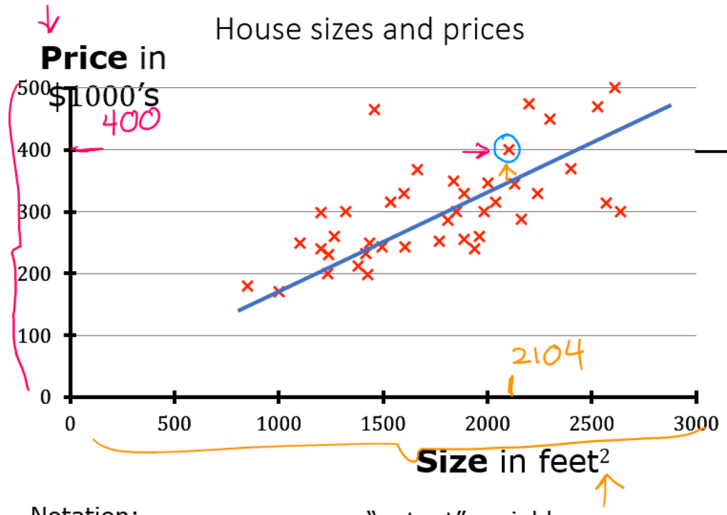
linear regression with one variable.

one X size

univariate linear regression.

one variable

DeepLearning.AI



Data table

(size in feet²)	price in $1000's
2104	400
1416	232
1534	315
852	178
...	...
3210	870

$(x^{(i)}, y^{(i)})$

$(x^{(i)}, y^{(i)})$ = i^{th} training example

index

(1^{st}, 2^{nd}, 3^{rd} ...)

Notation:
x = "input" variable
feature

y = "output" variable
"target" variable

m = number of training examples

DeepLearning.AI

input code useing jupyter

In [20]:
```
import numpy as up
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

In [21]:
```
x = np.array([11,12,13,14,15])
y = np.array([134.5,149,156,164,170])
```

In [27]:
```
linreg = LinearRegression()
```

In [28]:
```
x =  x.reshape(-1,1)
```

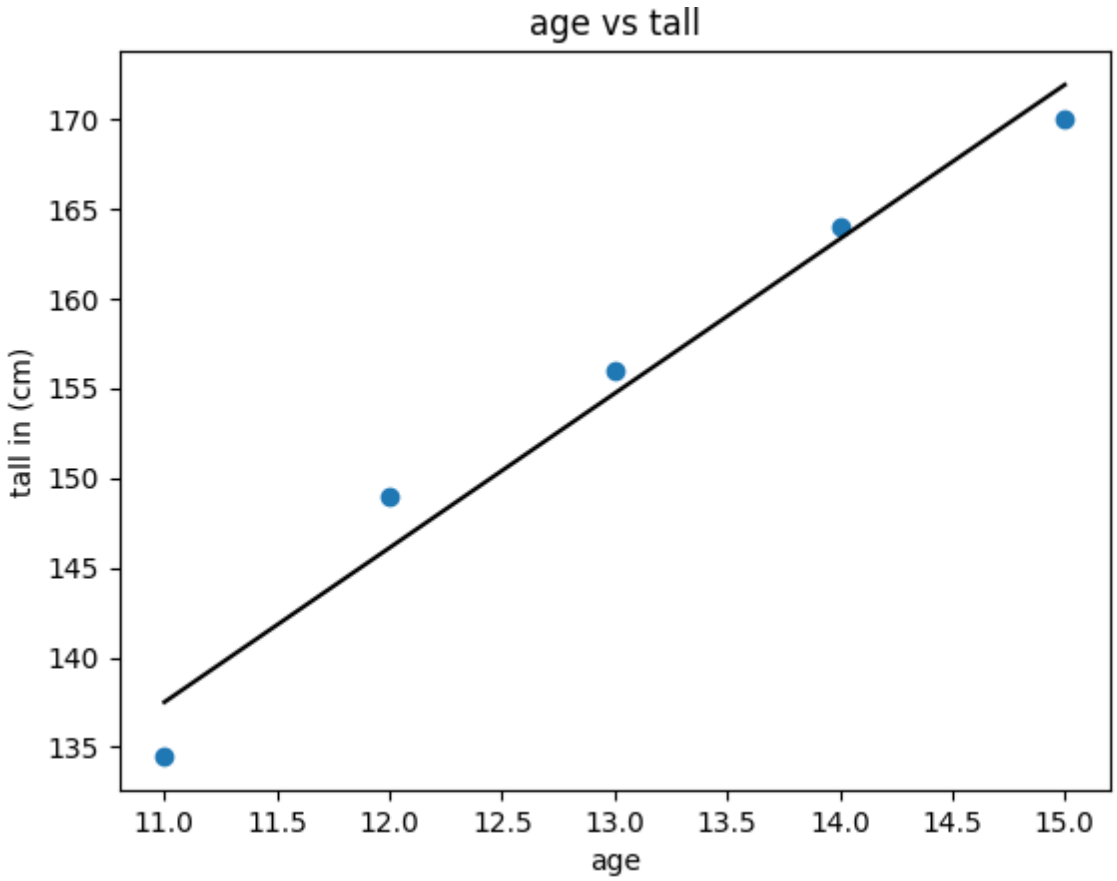
In [29]:
```
linreg.fit(x,y)
```

Out[29]:
```
LinearRegression
LinearRegression()
```

In [30]:
```
y_pred = linreg.predict(x)
```

In [33]:
```
plt.scatter(x,y)
plt.plot(x, y_pred, color='black')
plt.title('age vs tall')
plt.xlabel('age')
plt.ylabel('tall in (cm)')
plt.show()
```

output


age vs tall
tall in (cm)
170
165
160
155
150
145
140
135
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
age