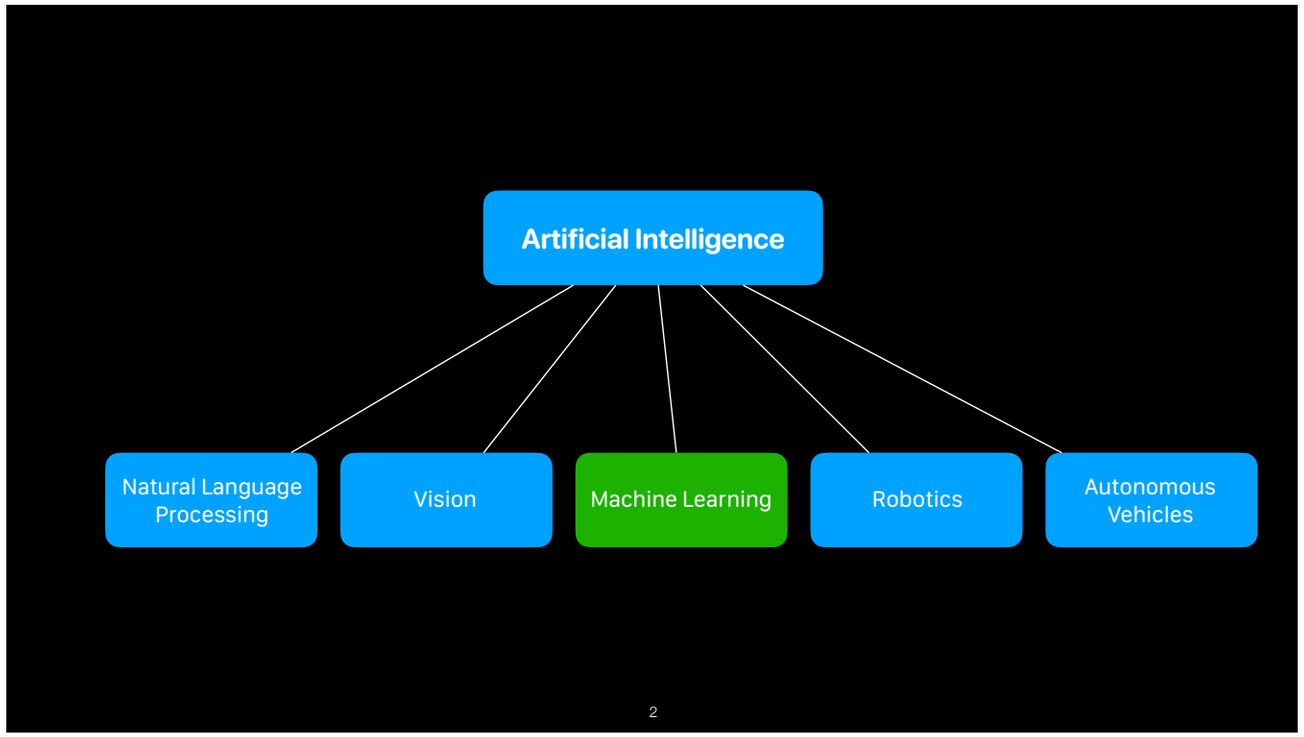
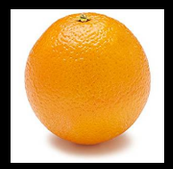
Machine Learning

An Introduction



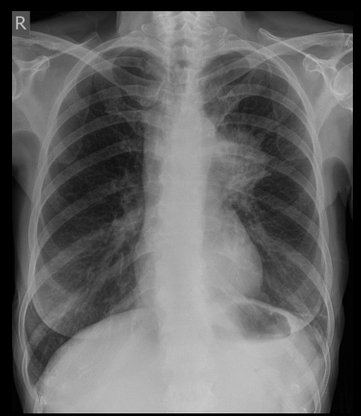
ML: target defined, machine gains experience by training itself to achieve target
NLP: speech and text recognition (email spam: text, Siri: speech) [providing customer service in banks]
Vision: enable machines to see and analyze, achieved also through ML
    Amazon Go, mobile apps, virtual baskets/carts, when takes an object machine adds, when placing back machine removes, without Facial recognition
    No cashiers, customers are allowed to leave and get an online receipt
Robotics: design and manufacture robots, perform tasks that are difficult for humans
AV: autopilot cars (Tesla), drones
ML is the core of all of these

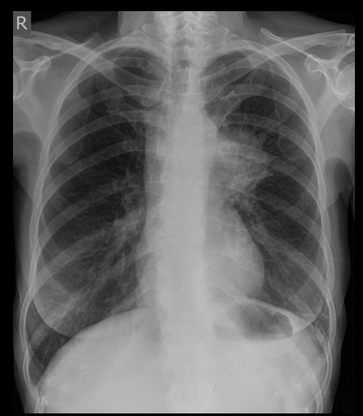
Why Machine Learning

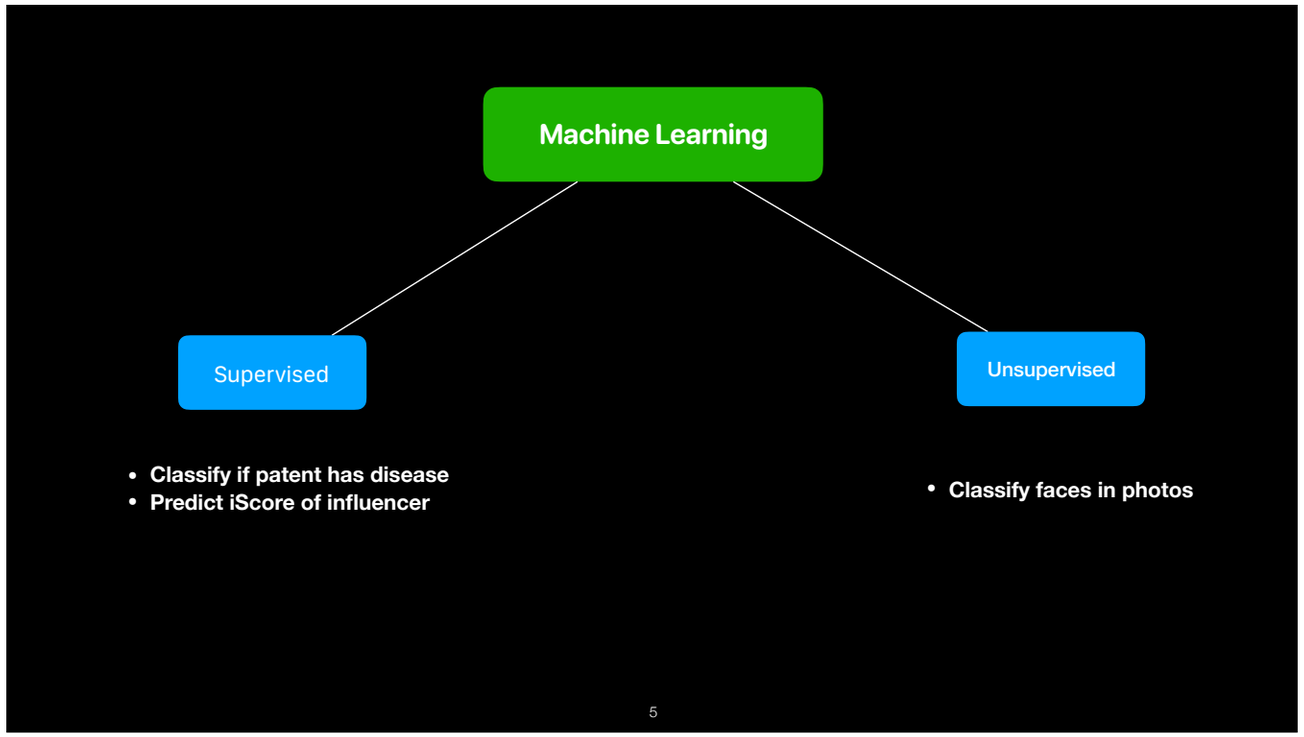





Why Machine Learning







Supervised: guessing from labeled/classified data, classify or predict a number
Unsupervised: bunch of photos of 6 people but without information about who is on which one and you want to divide this dataset into 6 piles, each with the photos of one individual. ‘ the way how iOS
reinforcement: teach machine to play chess, enemy AI in AAA games

Data Sets

Feature

Label/Class

Instance

Followers	Likes	Comments	iScore
15147	1084	2	6.523
13461	1889	4	9.483
7158	1720	43	4.567
14325	10747	10	7.652
13461	1889	4	7.129

70% Training Data

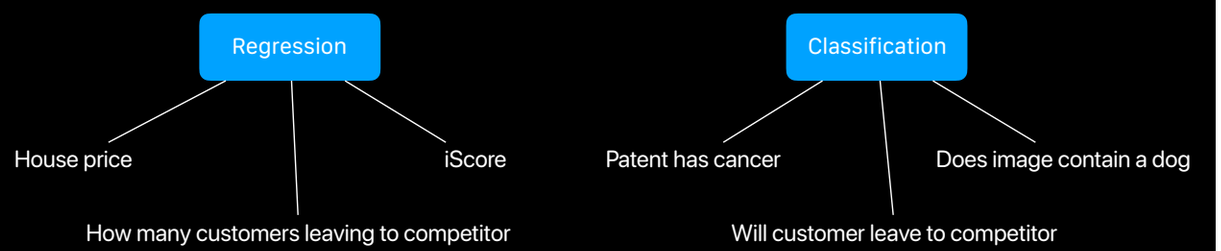
30% Testing Data

Supervised Learning

- Learn from correctly labeled data
- Data is fed to machine to train on
- Machine produces model used to predict new data
- Validate with testing set

we have access to examples of correct input-output pairs that we can show to the machine during the training phase
regression and classification: estimate/predict house price || does a given patient have cancer?

Supervised Learning

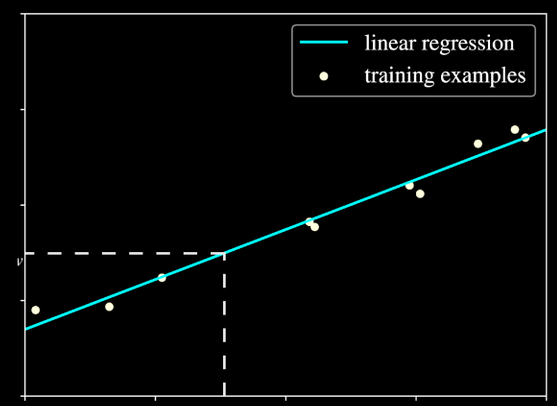


we have access to examples of correct input-output pairs that we can show to the machine during the training phase
regression and classification: estimate/predict house price || does a given patient have cancer?

Linear Regression

$$y = ax + b$$

- Build a model that predicts y for a given feature x
- Find the best values for a & b



for simplicity we have one feature here, but in real life it works on a big number of features

Solution

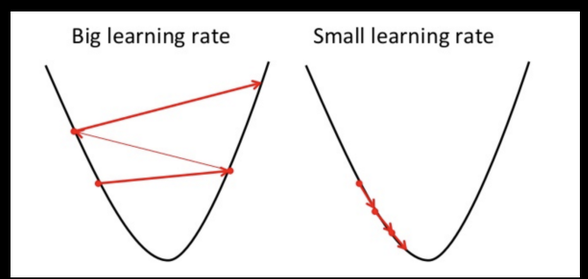
$$min\frac{1}{N}\sum_{1}^{N}(y - y_i)^2$$

- Solving a cost function (MSE)
- Gradient Descent

y: is the predicted value
yi real value

Gradient Descent

- Method of updating a & b to minimize the cost function
- Start with random values for a and b
- Change to reduce cost
- Values at minimum are used for the model



$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \implies \frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \cdot x_i \implies \frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

$$a_0 = a_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

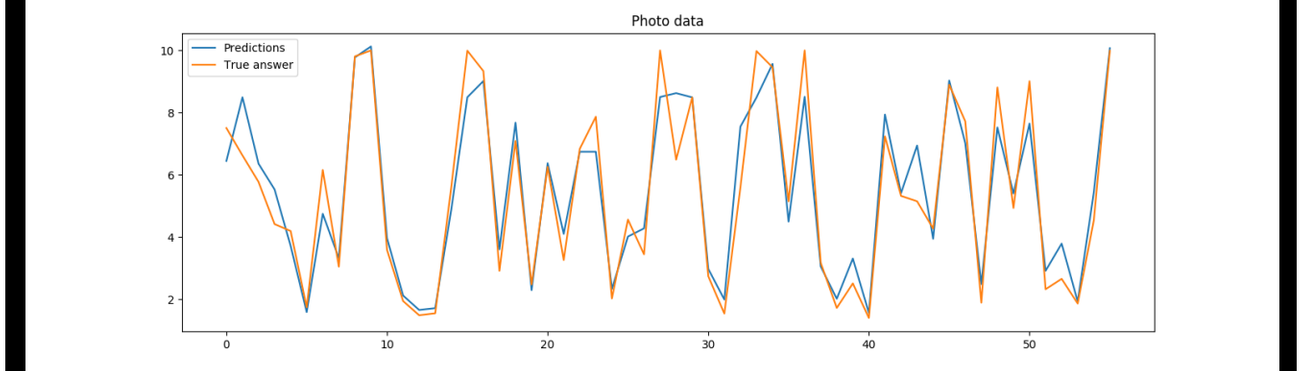
$$a_1 = a_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

nobody wants to know all the math behind this

Frameworks & Libraries

```
1  from sklearn.linear_model import LinearRegression
2  from sklearn.metrics import r2_score
3
4  clf = LinearRegression(normalize=True)
5  clf.fit(x_train,y_train)
6  y_pred = clf.predict(x_test)
7  print(r2_score(y_test,y_pred))
```

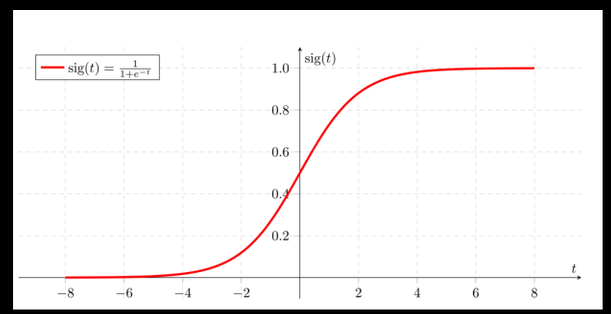
Experiment



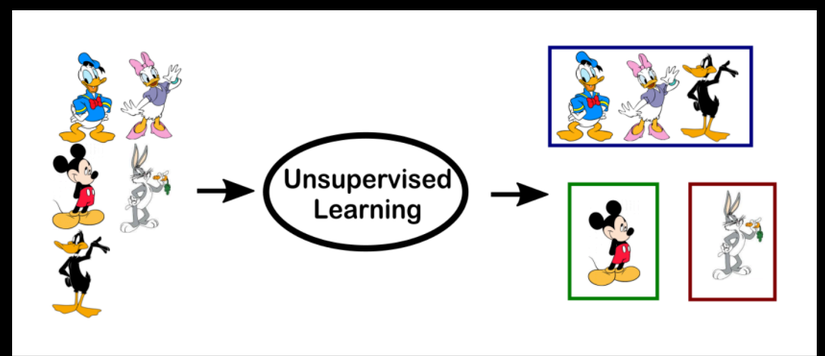
Classification

- Used when target (label) is categorical
- Eg: email is spam or not
- Sigmoid function
- Solve to minimize cost function using gradient descent

$$sig(t) = \frac{1}{1 + e^{-t}}$$



Unsupervised Learning



Unsupervised Learning

- Data is unlabelled
- Learn about the structure of our data
- Used for analysis and dimension reduction
- Will be covered in future presentations

“Thank you”

–Hussein