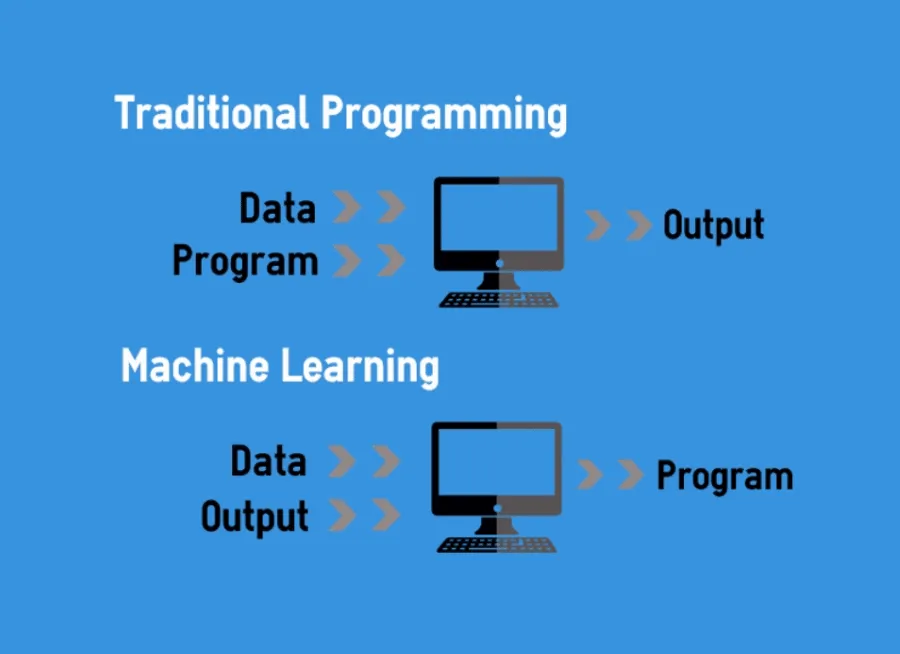
## Disposition

* Hvad er maskinlæring?

Hvad er maskinlæring/Machine Learning for en størrelse? Maskinlæring er en

utraditionel måde at skabe og køre programmer på. I modsætning til traditionel programmering, hvor vi giver vores computer et program og noget data til at køre dette program og give et output, så går maskinlæring ud på at vi giver computeren dataene, ligesom i traditionel programmering, men også et output frem for et program. I maskinlærings-programmering er det computeren selv der skal udvikle programmet, og det er der det bliver spændende. (Se evt. billede herunder)



* Hvad kan maskinlæring bruges til?

Maskinlæring kan bruges til mange mange ting. F.eks. kan vi bruge det til at se mønstre i data som den menneskelige psyke ville overse eller ikke opfange. Det kunne være ting som optimering af en kommunes økonomi eller forudse det næste store finansielle krak i økonomien. Mest relevant her i 2020, kan maskinlæring bruges til at kigge på en masse Covid-19 data og give os en bedre forståelse hvad vi kan have i vente.

* Hvad er klassifikation?
  + Decision stump

Decision stump kan nok bedst sammenlignes med det gamle brætspil “Hvem er hvem” for måden som at det virker på er gennem den samme Ja/Nej spørgsmåls metode som man bruger til at vinde spillet. Jo bedre spørgsmål, jo bedre classifier.

* + Nearest neighbour

Nearest neighbour, eller bedre kendt som KNN (K-Nearest neighbour), fungerer ved at vi kigger på vores data point og ser hvilken den er tættest på for at classify den. Har vi f.eks. 2 brackets; én der hedder “Lav IQ” med en base på 70 og en der hedder “Høj IQ” med en base på 150, så vil et nyt datapoint med en værdi på 90 ryge i Lav IQ puljen.

* Hvad er regression?
  + Lineær

Lineær regression er en maskinlæring algoritme baseret på overvåget læring. Regression modellerer en målt forudsigelsesværdi baseret på uafhængige variabler. Det bruges mest til at finde ud af sammenhængen mellem variabler og prognoser (forudsigelser). Forskellige regressionsmodeller adskiller sig efter - typen af forhold mellem afhængige og uafhængige variabler, de overvejer antallet af uafhængige variabler, der bruges.

* + Polynomial

Polynomial regression er en form for regressionsanalyse, hvor forholdet mellem den uafhængige variabel x og den afhængige variabel y er modelleret som en n-graders polynomium i x. Polynomial Regression passer til et ikke-lineært forhold mellem værdien af x og det tilsvarende betingede gennemsnit af y. Selvom polynomial regression passer til en ikke-lineær model af dataene, er det som et statistisk estimering problem lineært, i den forstand at regressions funktionen er lineær i de ukendte parametre, der estimeres ud fra dataene. Af denne grund betragtes polynomial regression som et specielt tilfælde af multipel lineær regression.

* + Neurale netværk

Neurale netværk kan reduceres til regressionsmodeller - et neuralt netværk kan "foregive" at være enhver form for regressionsmodel. For eksempel svarer dette meget enkle neurale netværk med kun en input neuron, en skjult neuron og en output neuron til en logistisk regression. Det tager flere afhængige variabler = inputparametre, multiplicerer dem med deres koefficienter (vægte) og kører dem igennem en sigmoid-aktiverings funktion og en enheds trins funktion, der meget ligner den logistiske regression funktion med dens fejl udtryk. Når dette neurale netværk trænes, vil det udføre gradient nedstigning for at finde koefficienter, der er bedre og passer til dataene, indtil det når frem til de optimale vægte for modellen.

* Projekt
  + Idébeskrivelse

Vores projekt går ud på at ændre en farve baseret på den farve vi siger ved brug af wekinator. Hvis vi siger “rød”, så skal farven rød komme frem. Siger vi farven “gul”, så skal farven gul komme frem, osv.

* + Teknisk problemstilling (se google sheet)

1. Hvordan sikres det at den ønskede farve vises, indtil der ønskes en ny
2. Hvordan sikres det at *Palles Hue* kun reagerer på reele voice commandoer.
3. Hvordan får vi bedst OSC-beskeder til at blive JavaScript output
   * Hvilken type algoritme har I valgt til jeres projekt, og hvorfor? Der skal gøres tydeligt rede for hvorfor, hvordan og på hvad Wekinator er blevet sat op og trænet, herunder især for valg af model og algoritme.

I dette projekt har vi benyttet algoritmen DTW (Dynamic Time Warping). Vi valgte DTW fordi at der **IKKE** skal skabes forskel på outputtet afhængigt af hvor hurtigt det blev sagt, og at vi var interesseret i hele tiden at lytte efter evt. input.

Da tanken var at lave et produkt ala *Phillips hue* med bonus af et smart-home feature som voice activation. Derfor blev stemme selvfølgelig sat op som input. Derfor blev Wekinator sat op med >50 samples pr. farve med udtalelse af den respektive farve. Denne proces kunne dog være effektiviseret ved at have benyttet flere menneskers stemme for en bedre model, **men det virkede ubarmhjertigt at få folk til at gennemgå samme smerte med dette, som jeg.** 😛

* + Der skal gøres tydeligt rede for hvordan det er hensigten at output skal opføre sig.

Hensigten med vores *“Palles Hue”* program var, at den via voice-commands som f.eks. “RØD”, skulle skifte til den respektive farve inden for regnbuens spektrum (+ sort & hvid).

* + Blokdiagram

