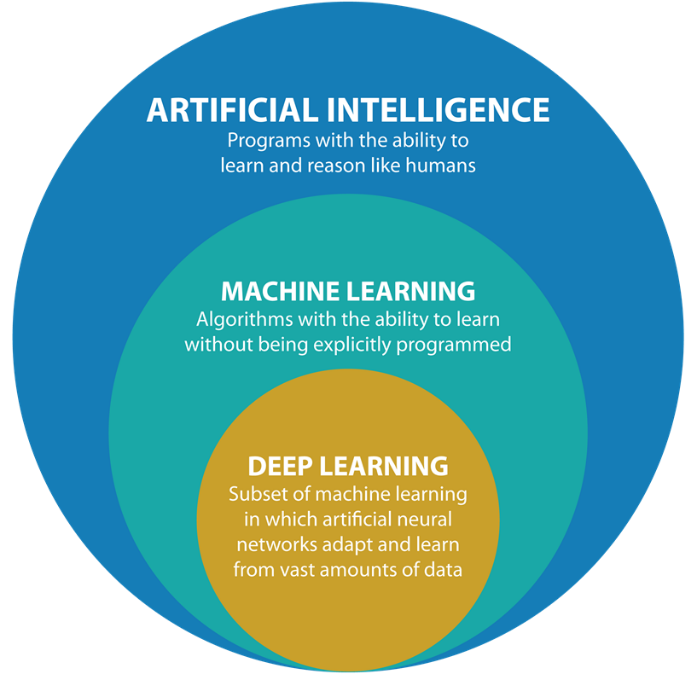
# BASICS START

Machine learning jedná se o algoritmy pro učení které přímo neprogramujeme. On nalezne patern z dat namísto toho aby mu byla přímo tento patern poskytnut.

Vytváří model s ohledem na sample datech (testovacích datech) za účelem predikování nebo rozhodovaní bez přímého zásahu programátora.



Pomocí strojového učení se řeší problémy jako předpověď počasí, ceny, rozpoznání obrázků apod.

Neprogramuje je přímo. Využíváme funkci pomocí které natrénujeme model, který následně využíváme při určování dalších dat.

* Učení s učitelem („supervised learning“) - pro vstupní data je určen správný výstup (třída pro klasifikaci nebo hodnota pro regresi).
* Učení bez učitele („unsupervised learning“) - ke vstupním datům není známý výstup.
* Kombinace učení s učitelem a bez učitele („semi-supervised learning“) - část vstupních dat je se známým výstupem, ale další data, typicky větší, jsou bez něj.
* Zpětnovazebné učení („reinforcement learning“), též učení posilováním, funguje na principu agenta, který interaguje s prostředím a za své akce dostává odměnu či trest. Agent se snaží maximalizovat odměnu. … umělá inteligence ve hrách samořiditelné drony apod.
* Klasifikace rozděluje data do dvou nebo několika tříd (učení s učitelem)
* Regrese odhaduje číselné hodnoty výstupu podle vstupu (učení s učitelem)
* Shlukování zařazuje objekty do skupin s podobnými vlastnostmi (učení bez učitele) … využití například na sociálních sítí
* stromy

Hledáme vztah mezi vstupními a hádaným parametrem. Častokrát nelze sehnat dokonalý model pouze se snažíme jenom zmenšit chybovost výsledné funkce. Strojové učení hledá funkci s nejmenší chybou, kterou pak můžeme uplatnit na vstupní parametry.

# Strojové učení - Příprava dat, Chyby v datech a bias, Korelace a kauzalita

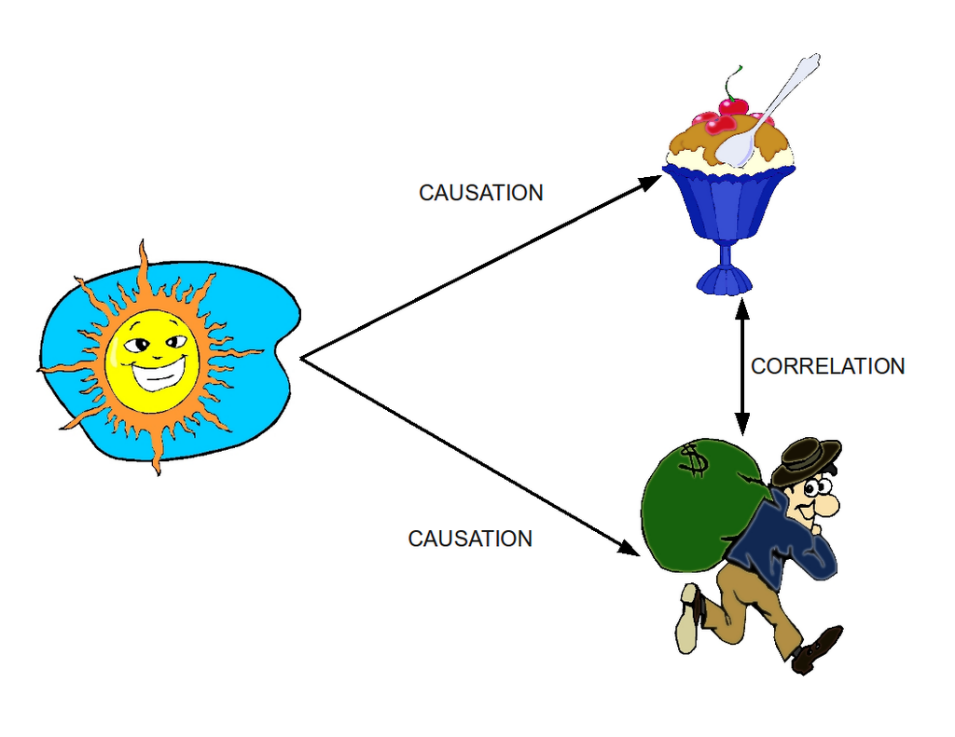
## Příprava dat

* Snažíme se se zbavit extrémních výjimek. Snažíme se co nejvíce zúžit rozptyl dat.
* Zbavit se záznamů s null hodnotami
* Všechny číselné hodnoty převést na hodnoty mezi 0 a 1 (některé ml modely [neuronové sítě] poskytují nejlepší výsledek právě když jsou vstupní hondoty v intervalu <0,1>)
* Upravit záznamy s nesmyslnými hodnotami nebo se jich zbavit
* Všechny parametry převést do nějakého standardizovaného stavu
* Zbavit se irelevantních atributů
* Atributy v data setu by měli být vyjádřeny pouze čísly

Korelace neimplikuje kauzalitu. Ale pro kauzalitu je nutná korelace.

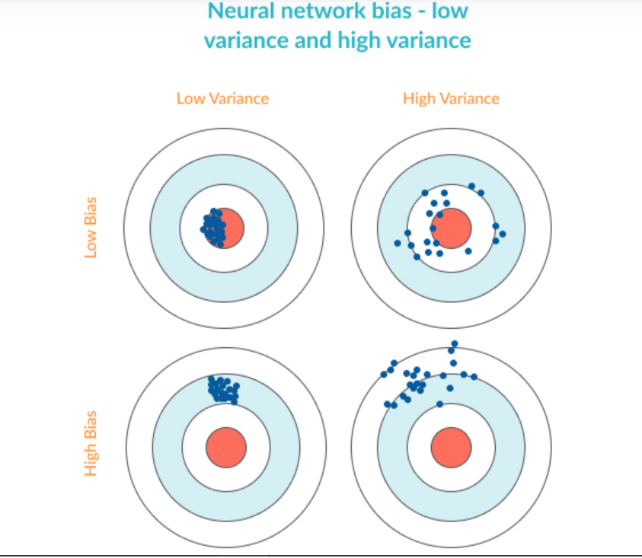
Korelace vztah mezi veličinami a procesy. Ve chvíli, kdy dojde ke změně v jedné věci dojde k změně v druhé věci. Jedna vílčina nezbytně neovlivňuje veličinu druhou.

V kauzalitě jedna veličina přímo ovlivňuje tu druhou. Protože se změnila jedna věc změní se i druhá.



Bias = jedná se o to, že máme nějaký extrém a funkce se podle něho řídí ale už se odchyluje od reality. Tím pádem máme dobrý model pro testovací data, ale v realitě model nebude zas tak dobrý.

Př: Prodáváme spíše dražší auta takže jsou ceny zaujaté aby byly vyšší protože v modelu není dostatek levnějších aut které by výslednou predikci vraceli do normálu a chybu (Bias) zmenšovali. Výsledkem je vyšší odhadovaná cena u levnějších aut.



# Strojové učení s využitím regrese a klasifikace

Rozdíl mezi regresním a klasifikačním problémem. (a): plnou čarou
je znázorněna regresní přímka, přerušovaná čára značí skutečnou funkční závislost Y na X. (b): rovina je rozdělena na dva regiony příslušné právě jedné třídě.
Plná čára ukazuje hranici mezi nimi.

Rozdíl mezi regresním a klasifikačním problémem. (a): plnou čarou je znázorněna regresní přímka, přerušovaná čára značí skutečnou funkční závislost Y na X. (b): rovina je rozdělena na dva regiony příslušné právě jedné třídě. Plná čára ukazuje hranici mezi nimi.

## Lineární regrese

* snažíme se proložit přímku mezi všemi body tak abychom dospěly k nejmenší
* opíráme se o square error tedy chybu na druhou z kterých následně uděláme průměr a zjistíme tak chybovost modelu
* výsledek lineární regrese je reálné číslo (hodnota odpovídající hodnotě na výsledné přímce) … oboje z from sklearn.metrics
* pomocí algoritmu Stochastic Gradient Descent (SGD) pro nalezení adekvátní funkce, nude mí minimální kvadratickou chybu vůči všem bodům v souřadnicovém systému

Obsah obrázku diagram

Popis byl vytvořen automaticky

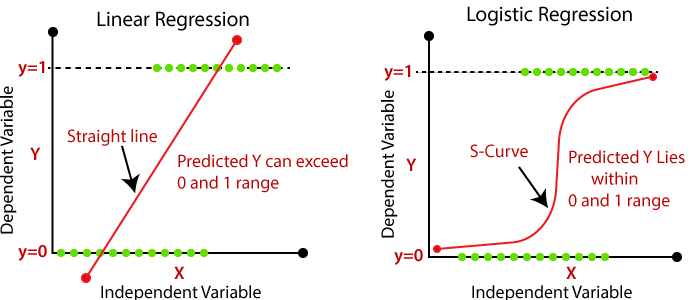
mae = mean\_absolute\_error(y\_test, y\_pred)

0.6504544928096602

mse = mean\_squared\_error(y\_test, y\_pred, squared=False)

0.9384014413084975

<https://colab.research.google.com/drive/1mOroDym6F0vWQYk_qWma0j5KsgOQHJas?usp=sharing>



## Klasifikace

* logistická regrese převádí regresi na klasifikaci pomocí sigmoidy
* snažím se do grafu vložit přímku tak abych rozdělil co nejlépe dvě množiny prvků, následně funkce zjisti, do jaké kategorie hledaný prvek patří, jestli do kategorie na jedné straně od přímky nebo na druhé straně
* výsledek klasifikace je nějaká budě accuracy\_score(y\_test, y\_pred) … (např.: 0.8, 0.9, 1.1) nebo classification\_report(y\_test, y\_pred) … oboje from sklearn.metrics

precision recall f1-score support

0 1.00 1.00 1.00 8

1 0.83 0.91 0.87 11

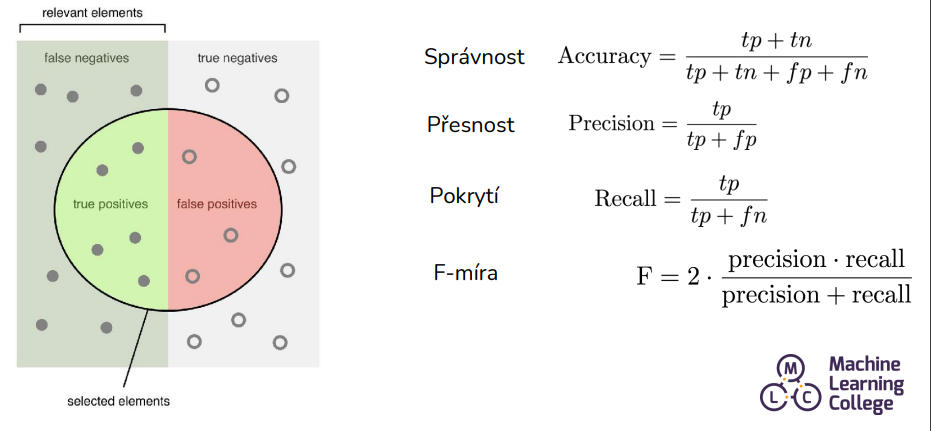
2 0.90 0.82 0.86 11

accuracy 0.90 30

macro avg 0.91 0.91 0.91 30

weighted avg 0.90 0.90 0.90 30

<https://colab.research.google.com/drive/1J2j72dMF1q0Errgu2RkuMCOZUYoSiIBd?usp=sharing>



## Rozhodovací stromy

Jsou robustnější na rozdíl od funkcí nemusí vždy zachytávat tu střední hodnotu.

Obsah obrázku diagram

Popis byl vytvořen automaticky

Klastrová analýza (**Cluster analysis**)

Má za úkol shlukovat do skupin objekty takovým způsobem že objekty ve stejné skupině(clusteru) jsou si podobní samy sobě více než objekty v jiných skupinách(clusterech).

Obsah obrázku tabulka

Popis byl vytvořen automaticky

# Strojové učení s využitím umělých neornových sítí

<https://colab.research.google.com/drive/1XDJuMFQyvx5vTVE5E4BQCZwLbVs1YJGa?usp=sharing>

Problémy jsou složitější, něž co zvládne vyřešit lineární regrese. Pomocí neuronových sítí zvládneme vygenerovat složitější konstrukce.

Základní stavební kámen je neuron. Každý neuron představuje nějakou funkci, která částečně pokrývá testovací data. Funkce v neuronech se navzájem překrývají a pokud jsou správně rozmístěny mohou pokrýt testovací data dohromady.

Vytvořit takový model potřebuje mnohem více výpočetní síly a pro hodně případů se jedná o overkill.

Obsah obrázku text, interiér, osoba, černá tabule

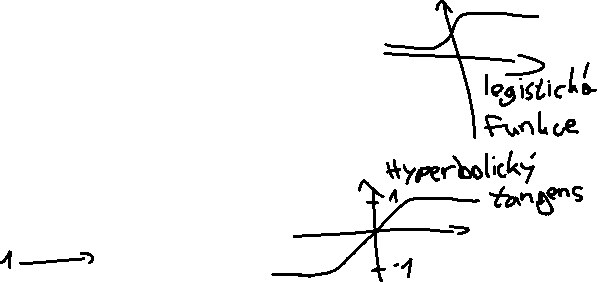
Popis byl vytvořen automaticky

Perceptron = Neuronová síť obsahuje jen jediný neuron.

* N vstupů a jeden výstup. Vezme každý z těch vstupů a pronásobí to s nějakou váhou. Výsledky sečte.
* 1 která se násobí s vahou w0 … Bias který realizuje tedy lineární posuv.
* Výsledné číslo se ještě prožene aktivační funkcí. Nějaká nelineární funkce (v jednodušších případech lineární funkce), či logistická funkce. Pokud jako aktivační funkci použijeme logistikou funkci (sigmoid) dostaneme logistická regresi. Pokud bychom použili lineární funkci jako aktivační funkci dostali bychom lineární regresi.
* TANH (hyperbolický tangens) lze využít, když výstup potřebujeme dostat do další funkce kde hodnota musí být normalizovaná tedy v intervalu od -1 do 1.
* Za běhu se algoritmus bude snažit upravovat váhy tak aby výsledek byl co nejpodobnější trénovacím příkladům (chyba co nejmenší).

Obsah obrázku diagram

Popis byl vytvořen automaticky



Algoritmus zpětné propagace chyby.

Abychom mohli spočítat úpravu vah u jednotlivých neuron nejprve musíme provést krok vpřed za pomocí nějakých sample (testovacích) dat provedeme s vahami které na začátku byly přiřazeny náhodně. Dostaneme výstupní hodnotu. Výstupní hodnotu porovnáme s výslednou hodnotou z sample (testovacích) dat a spočítáme jak bychom měli opravit váhy aby jsme tu vypočtenou chybu zmenšili. Nejprve upravujeme váhy v první vrstvě (laier) od výstupního neuron tím zas vynikne nová chyba kterou budeme upravovat v následující vrstvě (laier) a postupně se vracíme zpátky až do vrstvy u vstupních neuronů.

Tedy propagujeme chybu, která nám vznikla na výstupu na vstup.  
Deep learning se říká neuronové síti s velkém množství neuronů a vrstev s nimi. Využívá velký výpočetní výkon. Používá se jiné aktivační funkce či algoritmy oprav.