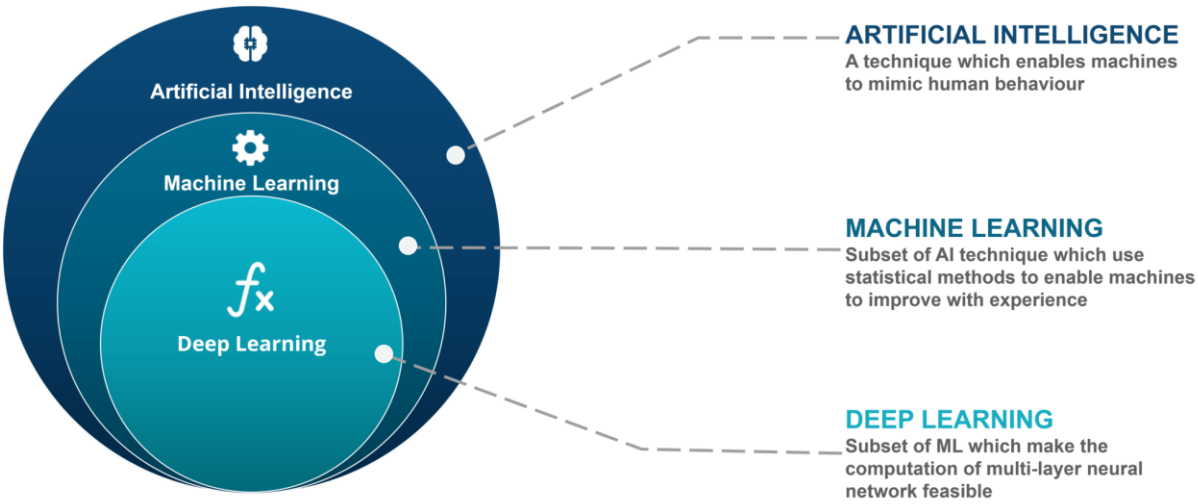
Strojové a hluboké učení v DPZ

Oblast dálkového průzkumu Země prošla revolucí díky výpočetnímu výkonu. Konkrétně naše schopnost, resp. schopnost počítačů, automaticky zpracovávat, klasifikovat a interpretovat obrovské množství satelitních a leteckých snímků vychází z technik vyvinutých v rámci umělé inteligence (AI). Tyto techniky spadají pod tzv. strojové a hluboké učení. Následující diagram znázorňuje, jak umělá inteligence a strojové a hluboké učení spolu souvisí.



Porovnání tradičních statistických modelů s modely strojového učení

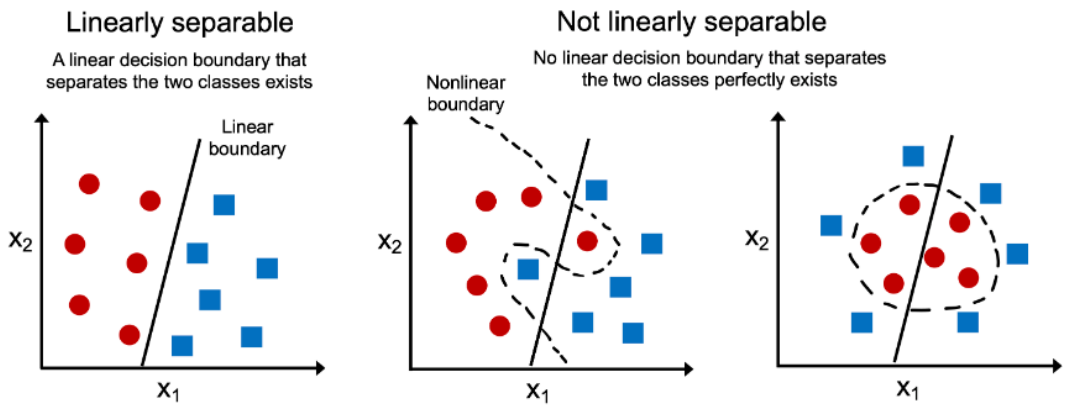
Statistické modely

- Statistické modely jsou často parametrické a předpokládají, že data následují nějaké konkrétní matematické rozdělení (např. Normální rozdělení)
- Jsou velmi závislé na těchto předpokladech a fungují jen tehdy, pokud jsou tyto předpoklady splněny
- Fungují dobře na menších strukturovaných datasetech, kde jsou vztahy poměrně jednoduché (**lineární problémy**)
- Obtížně kombinují spojitá a kategorická data

Modely strojového učení

- Modely strojového učení jsou naopak často neparametrické (struktura modelu je flexibilní a je dána samotnými daty)
- Nízká závislost na explicitních předpokladech o základním rozložení dat (důraz je kladen na empirický výkon)
- Mají vynikající schopnost zpracovávat velmi velké, komplexní a vysokorozměrné datasety (např. multispektrální obrazová data)
- Řeší složité nelineární problémy
- Nevýhodou je, že na rozdíl od tradičních statistických modelů často fungují jako tzv. „černé skříňky“, což znamená, že nejsme schopni plně vysvětlit komplexitu těchto modelů (**platí zejména u modelů hlubokého učení**)

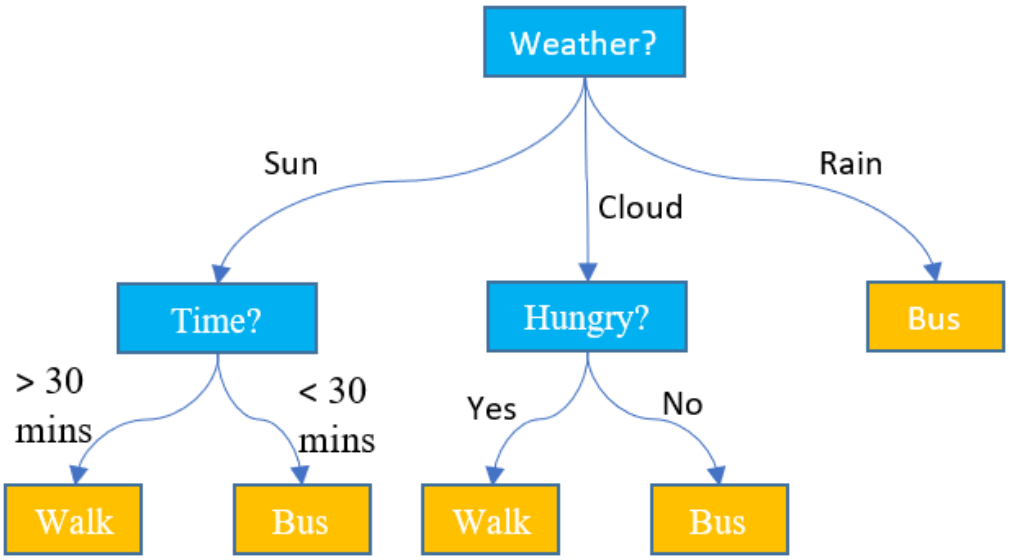
Příklady lineárních a nelineárních problémů:



Příklad klasifikátoru strojového učení – Random Forest

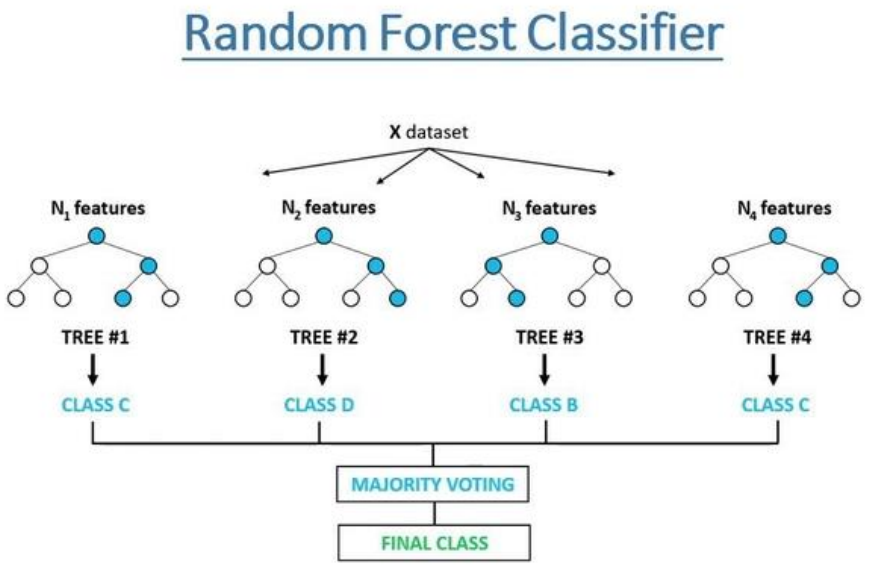
Random Forest je jedním z nejrozšířenějších a nejvyužívanějších modelů strojového učení nejen v oblasti DPZ. Jedná se o metodu řízené klasifikace, kdy se model učí vlastnosti a vztahy v trénovacích datech, které pak aplikuje na celý dataset. Vychází z tzv. rozhodovacích stromů, resp. jedná se o víc rozhodovacích stromů dohromady, které tvoří tzv. ansámbl (*ensemble*). V porovnání s jednotlivými rozhodovacími stromy jsou výsledky random forest méně zatížené zkreslením a jsou proto spolehlivější a méně náchylné na nepřesnosti v trénovacích datech.

- **rozhodovací strom** – jedná se o neparametrický algoritmus řízeného učení, který svou strukturou připomíná podobu vývojového diagramu
- **ansámbl** – skupina nezávislých modelů (v případě random forest skupina rozhodovacích stromů), které dohromady tvoří jeden silný a robustní model (klasifikátor)



Ukázka rozhodovacího stromu

Random forest tedy kombinuje více rozhodovacích stromů dohromady. Tyto stromy jsou na sobě navíc nezávislé, jelikož do každého stromu při trénování vstupují trochu jiná a náhodná data z referenčního datasetu. Tato náhodnost je dvojího charakteru. Buď můžeme mít náhodnost ve vstupních vzorcích (tj. různá referenční data pro různé stromy) nebo je náhodnost ve vstupních příznacích (tj. do jednotlivých stromů vstupují např. různá pásma z multispektrálních dat, pokud mluvíme o DPZ). Jednotlivé stromy poté pomocí vstupních příznaků rozdělují vstupní referenční vzorky na dvě co „nejčistší“ části. Toto rozdělování trvá tak dlouho, dokud má strom možnost se rozrůstat (větvit). Rozrůstání stromu je řízeno tzv. hyperparametry, které jsou před začátkem trénování nastaveny uživatelem. Při klasifikaci poté jednotlivé stromy zařazují dané vzorky do konkrétních tříd. Výslednou třídou je ta třída, která dostala celkově nejvíce hlasů (tj. danou třídu určilo nejvíce stromů). Díky tomu je random forest klasifikátor poměrně dost odolný vůči chybám v trénovacím datasetu.



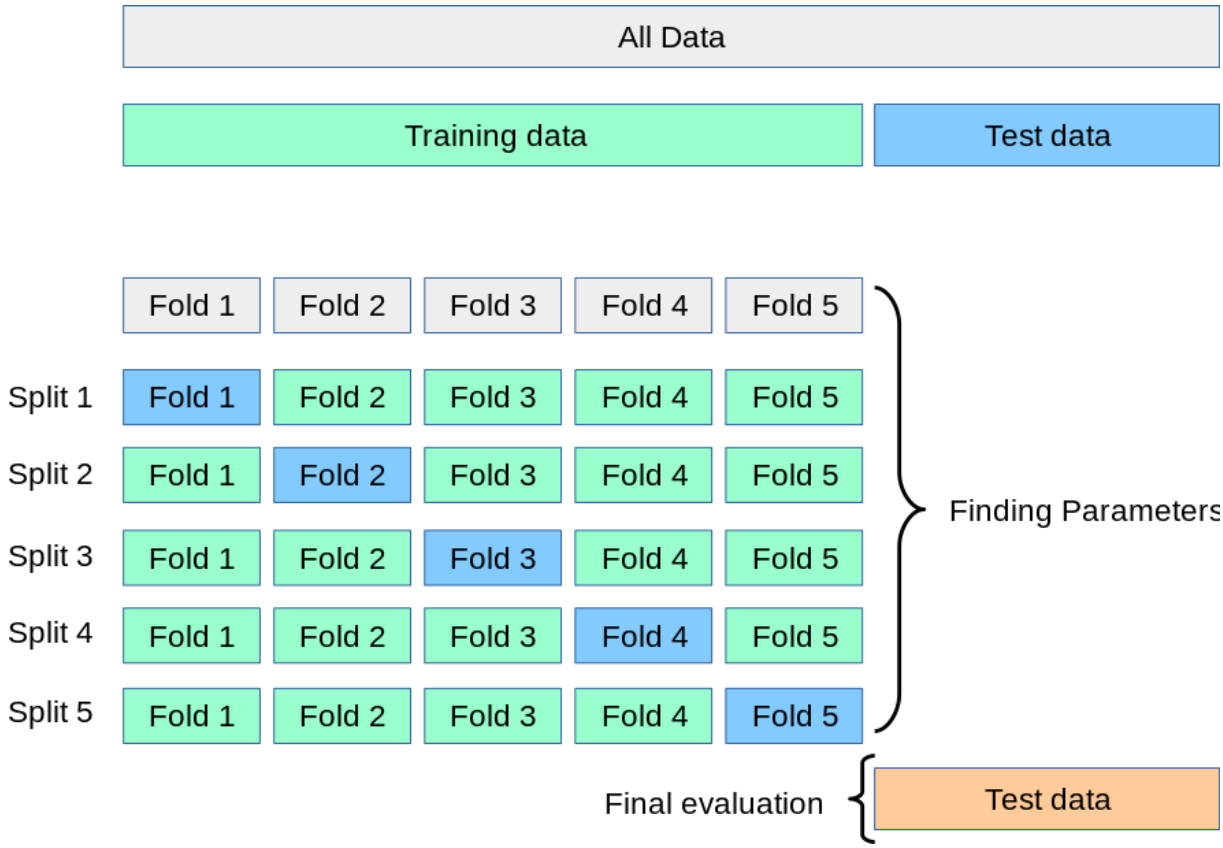
Hyperparametry random forest

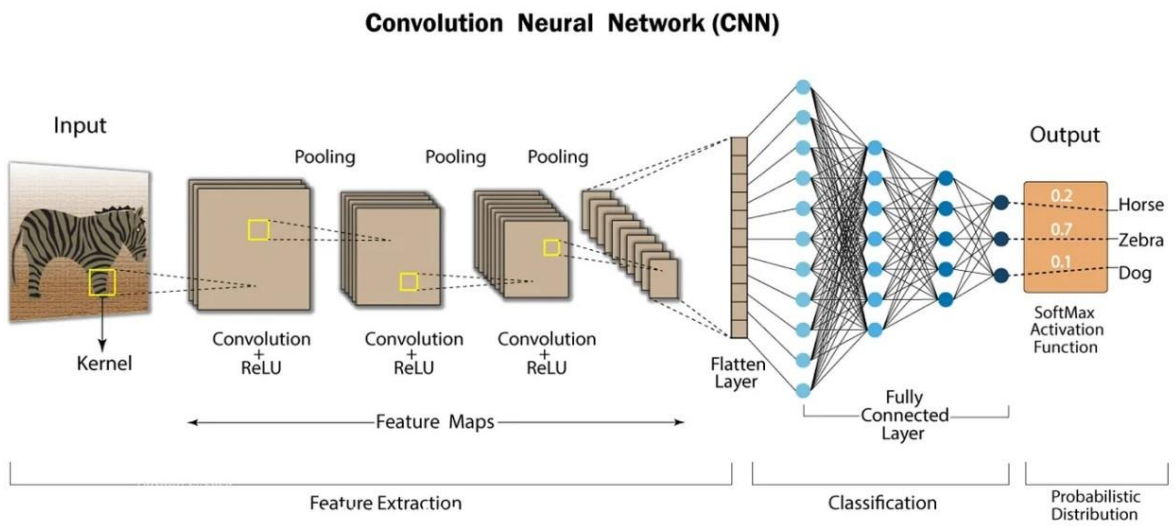
- počet stromů (***n_estimators***) – celkový počet rozhodovacích stromů
- maximální hloubka (***max_depth***) – maximální hloubka, do které se mohou stromy rozrůst (lze nastavit i na neomezené rozrůstání)
- minimální počet vzorků potřebných k rozdělení uzlu (***min_samples_split***) – jinými slovy kolik je potřeba mít v uzlu alespoň vzorků, aby jej bylo možné rozdělit na další dva uzly
- minimální počet vzorků v koncovém uzlu (listu) (***min_samples_leaf***) – jinými slovy kolik je potřeba mít v koncovém uzlu alespoň vzorků, aby daný uzel mohl být koncový (pokud tato podmínka není splněna, tak není možné předchozí uzel dále dělit)

Správným nastavením hyperparametrů (***max_depth***, ***min_samples_split***, ***min_samples_leaf***) se předejde tzv. přetrénování (***overfitting***), tj. situaci, kdy se model až příliš dobře naučil fungovat na trénovacích datech, ale na datech, které doposud neviděl, funguje o poznání hůře či vůbec. Proto je potřeba model vždy vyzkoušet jak na trénovacích, tak i na testovacích (validačních) datech.

Hledání optimálních hyperparametrů ovšem není triviální záležitostí. V ideálním případě je potřeba vyzkoušet co nejvíce kombinací hyperparametrů. Manuální zkoušení různých kombinací je ovšem neefektivní a zejména časově velmi náročné. Proto existují metody, které to udělají za člověka automaticky. Jedná se o tzv. ***grid_search*** a ***random_search***. Ty fungují tak, že jim poskytneme hyperparametry a jejich hodnoty, které chceme testovat. V případě *grid_search* jsou poté testovány všechny možné kombinace, v případě *random_search* jen nějaký jejich náhodný subset. Toto hledání optimálních hyperparametrů probíhá na trénovacích datech nejčastěji pomocí tzv. **křížové validace** (*cross validation*).

Křížová validace je proces, kdy jsou trénovací data rozdělena na *k* menších částí. Jedna část je vždy ponechána stranou, zatímco na zbylých *k-1* částí je model trénován. Část daná stranou je následně použita k validování. Pro jedno nastavení hyperparametrů je toto provedeno tolikrát, na kolik částí byl trénovací dataset rozdělen (*k-krát*). Trénovací přesnost je poté určena jako průměr přesností dosažených v jednotlivých *k* bězích. Celý tento proces je aplikován na všechny možné kombinace hyperparametrů. Jedná se tedy časově i výpočetně o nejnáročnější část a ideálně je k tomu zapotřebí výkonný počítač s několika jádry. V opačném případě může tento proces trvat až příliš dlouho. Výsledkem je kombinace hyperparametrů, které daly nejvyšší přesnost. Tato kombinace hyperparametrů je nakonec ještě otestována testovacími daty, které byly dány stranou před celým procesem křížové validace stranou. Nejčastějšími metrikami používanými ke zjišťování přesnosti modelu jsou *F1-score, precission, recall,* či celková přesnost (*overall accuracy*). Křížovou validaci lze aplikovat na jakýkoliv model strojového učení, u kterého je potřeba najít optimální nastavení.



Konvoluční neuronové sítě

V současné době jsou velmi populární takzvané konvoluční neuronové sítě patřící pod hluboké učení. Svou oblíbenost získali díky dosahování vysokých přesností v úkolech založených na obrazových datech. Velmi efektivní jsou například v detekci prostorových anomálií. Jejich velkou výhodou oproti klasickým algoritmům je, že kromě samotných pixelů dokážou pracovat i s jejich okolím. Díky tomu jsou tak schopny „vnímat“ i prostorový kontext jednotlivých prvků v obrazu, což klasické algoritmy nedokážou. Jak jejich název napovídá, princip konvolučních sítí je založen na konvolučních vrstvách, které na obrazová data aplikují filtry, pomocí kterých detekují vzory v datech, které se učí z trénovacích množin. Nevýhodou ovšem je, že vyžadují poměrně velké množství trénovacích dat, která navíc musí být ve formě kontinuální vrstvy bez mezer (nestačí tedy mít jen několik bodů či polygonů, ale je potřeba mít označenou větší bezešvou plochu). Zároveň jsou i velmi náročné na výpočetní zdroje, a bez výkonné grafické karty je často nemožné je úspěšně natrénovat.