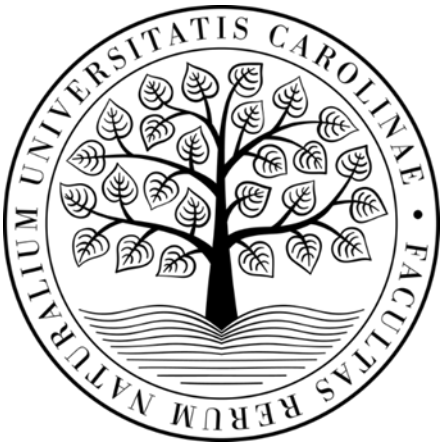
Univerzita Karlova

Přírodovědecká fakulta

Studijní program: Geoinformatika, kartografie a dálkový průzkum Země



Karolína Drápelová, Anna Wildová

1. ročník studijního programu Geoinformatika, kartografie a dálkový průzkum Země

Strojové učení: generalizace modelu

Geoinformatika

Akademický rok 2025/2026

Praha, 2025

Zadání:

1. Pořízení a příprava datového souboru

Pro zvolenou scénu dat Sentinel-2:

- vyberte alespoň 3 třídy krajinného pokryvu (např. les, voda, zemědělská půd půda),
- proveďte sběr referenčních polygonů reprezentujících jednotlivé třídy – trénovací a testovací,
- z dat odvoďte vstupní atributy (např. spektrální pásma, vegetační indexy).

2. Návrh a trénování modelu Decision Tree

V programovacím jazyce Python:

- natrénujte klasifikační Decision Tree,
- experimentujte s různými parametry stromu, zejména:
    - maximální hloubka stromu,
    - minimální počet vzorků v listu,
- posuďte vliv těchto parametrů na generalizační schopnost modelu.

3. Vyhodnocení a výstupy cvičení

Výstupem cvičení bude:

a. Vizualizace vstupních dat

- scatterplot vstupních dat (příznaků),

b. Schematický popis algoritmu Decision Tree

- princip rekurzivního dělení prostoru,
- vysvětlení použitých kritérií dělení,

c. Diskuse výsledků

- zhodnocení generalizační schopnosti modelu,
- vliv parametrů stromu na přeučení a stabilitu klasifikace,
- porovnání výsledků pro různé konfigurace Decision Tree.

Vypracování:

1. **Pořízení a příprava datového souboru**

   Z Copernicus Browser byla vybrána scéna 12. 8. 2025 pokrývající střed ČR a sever Rakouska (*L2A_T33UWQ_A044049_20250812T100213*). V gisovém softwaru byl poté vytvořen rastr zahrnující NDVI hodnoty pro celou scénu a všechna pásma byla spojena do jednoho souboru. Poté byly vytvořeny polygony land coveru – konkrétně voda, les a zástavba – jak pro trénovací data, tak testovací. Polygonové vrstvy byly převedeny do rastrů, aby se s nimi dalo pracovat v pythonu s knihovnou *rasterio*.

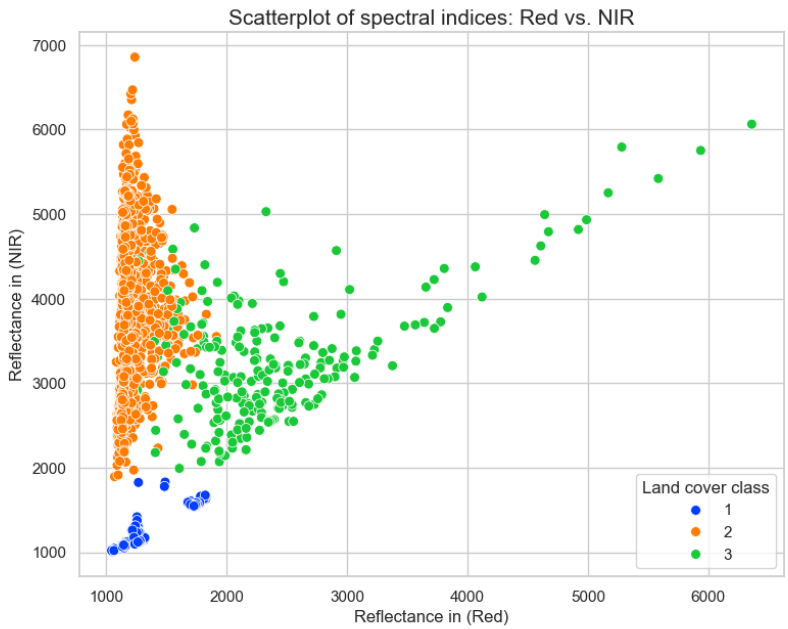
2. **Návrh a trénování modelu Decision Tree**

   Pro zpracování v pythonu byly využity knihovny *pandas, seaborn, sklearn, matplotlib* a *rasterio*. Data byla načtena pomocí knihovny *rasterio*, knihovny *seaborn a matplotlib* byly využity pro vizualizaci scatterplotu a *sklearn* pro samotný model DecisionTree. U modelu byly měněny parametry *max_depth a min_samples_leaf*.

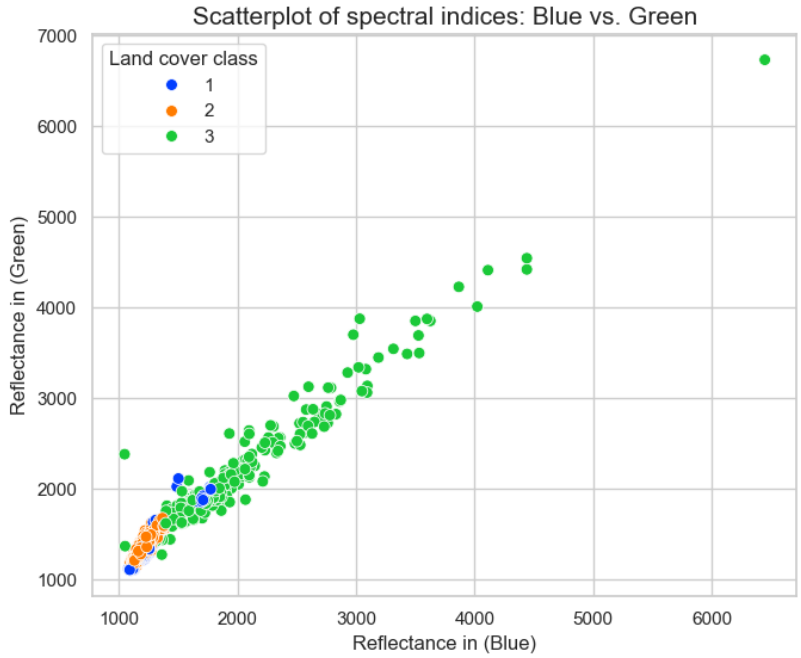
3. **Vyhodnocení a výstupy cvičení**

   a)

   Na obrázcích 1 a 2 je scatterplot příznaků, konkrétně pro červené a blízké infračervené pásmo a modré a zelené pásmo. Na obrázku 1 lze jasně rozlišit různé typy land coveru, vodní plochy se vyznačují nízkou odrazivostí v obou pásmech (modré body), lesy se vyznačují velkou odrazivostí v NIR pásmu (oranžové body), zástavba poté leží na diagonále (zelené body). Na obrázku 2 nelze rozlišit jednotlivé typy land coveru, všechny leží na diagonále.



Obrázek 1: Scatterplot Red x NIR (1: voda, 2: les, 3: zástavba).

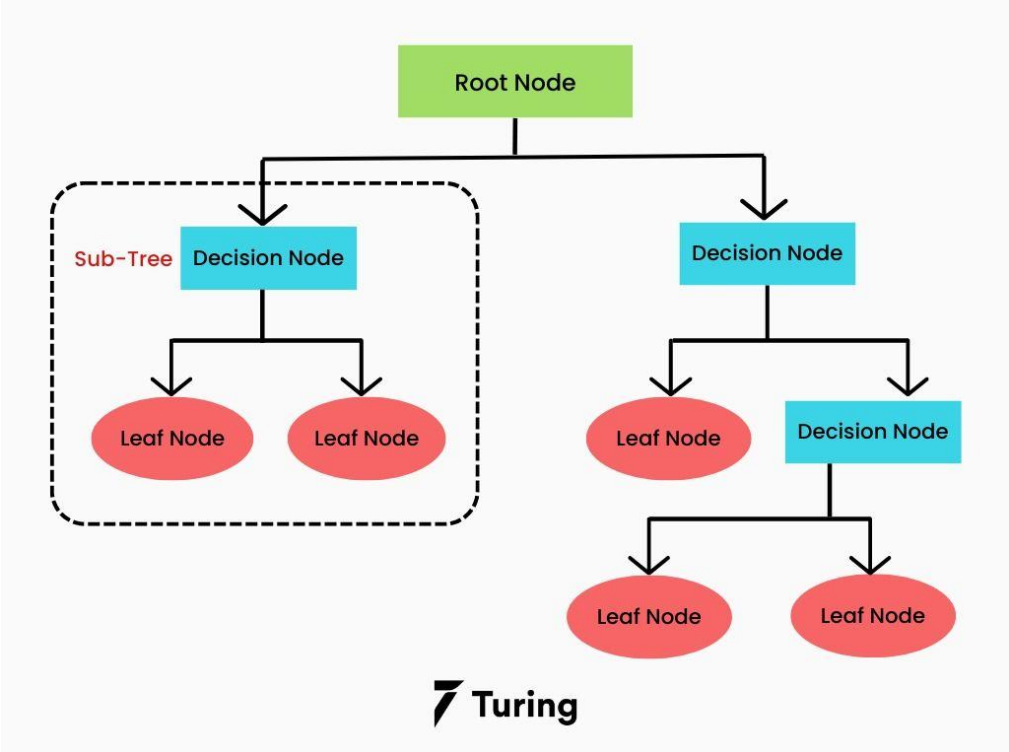


Obrázek 2: Scatterplot Blue x Green (1: voda, 2: les, 3: zástavba).

b)

Rozhodovací strom (Decision Tree) je metoda strojového učení, která využívá stromovou strukturu k modelování rozhodování a predikci cílových hodnot. Strom se skládá ze tří hlavních typů uzlů:

- **Kořenový uzel (Root Node):** Představuje celý datový soubor na začátku procesu.
- **Vnitřní uzly (Internal/Decision Nodes):** Reprezentují testování konkrétního atributu (např. *NDVI > 0,2?*). Zde dochází k větvení.
- **Listové uzly (Leaf Nodes):** Koncové body, které nesou finální výsledek (klasifikaci třídy, např. *les* nebo *voda*).



Obrázek 3: Schéma Decision Tree (Turing, 2022).

Princip rekurzivního dělení prostoru: Algoritmus funguje na principu rekurzivního dělení. Proces začíná v kořenovém uzlu a opakovaně dělí data do menších podskupin na základě hodnot vstupních atributů:

- Algoritmus vybere atribut, který nejlépe odděluje data.
- Vytvoří větve pro různé hodnoty tohoto atributu.
- Tento postup se rekurzivně opakuje pro každou vzniklou podskupinu, dokud nejsou splněna „stopping“ kritéria (např. dosažení maximální hloubky stromu).

Vysvětlení použitých kritérií dělení: Klíčovým krokem je určení, podle čeho a kde data rozdělit. Algoritmus vybírá rozdělení, které maximalizuje homogenitu vzniklých uzlů. V knihovně Scikit-learn nejčastěji využívají tyto metriky:

- Gini Impurity (Giniho index): Měří pravděpodobnost nesprávné klasifikace náhodně vybraného prvku. Cílem je minimalizovat tuto hodnotu (hodnota 0 znamená dokonale čistý uzel obsahující pouze jednu třídu).
- Entropy (Entropie) a Information Gain: Měří míru neurčitosti v datech. Rozdělení s nejvyšším informačním ziskem (Information Gain) nejvíce snižuje entropii, a tedy nejlépe uspořádává data.

c)

Generalizační schopnost modelu byla posouzena porovnáním přesnosti na trénovací a testovací sadě (tabulka 1).

Podučení (Underfitting): K podučení dochází pouze u velmi mělkého stromu (hloubka 1). Zde je přesnost na trénovacích datech přibližně 88 %, ale na testovacích datech jen 65 %. Model v tomto nastavení nedokáže dostatečně oddělit třídy.

Optimální nastavení: Nejvyšší přesnost na testovacích datech (83,75 %) byla dosažena již při hloubce 4. Další zvyšování hloubky již nevedlo k výraznému zlepšení klasifikace na testovací sadě, což naznačuje, že hlavní spektrální vzory byly odhaleny již v prvních úrovních stromu.

Přeučení (Overfitting): Při nastavení min_samples_leaf = 1 a vysoké hloubce (např. 25) dosáhla trénovací přesnost téměř maxima (99,87 %), zatímco testovací přesnost klesla na 81,48 %. Rozdíl cca 18procentních bodů potvrzuje, že se model začal učit šum v trénovacích datech.

Vliv parametrů stromu na stabilitu klasifikace:

Zvyšování hloubky z 1 na 2 přineslo největší skok v přesnosti (z 65 % na 83 % na testovacích datech). Po překročení hloubky 4 se testovací přesnost ustálila a v některých případech (u nízkého min_samples_leaf) začala mírně klesat v důsledku přeučení.

Minimální počet vzorků v listu se ukázal jako klíčový pro stabilitu modelu při vyšších hloubkách.

Nízké hodnoty (1) - umožnily modelu dosáhnout extrémní trénovací přesnosti (téměř 100 %), ale vedly k nestabilním výsledkům na testovací sadě (pokles k 81,5 %).

Vyšší hodnoty (např. 50) – fungovaly proti přeučení. Při hloubce 25 a min_samples_leaf = 50 se trénovací přesnost zastavila na 98,99 % a testovací přesnost zůstala stabilní na 83,05 %. Model s tímto nastavením je sice o něco méně přesný než maximum při hloubce 4, ale je méně náchylný k výkyvům.

P. listů	Accuracy	Hloubka 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	25
1	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9896	0,9905	0,9914	0,9919	0,9926	0,9932	0,9939	0,9944	0,995	0,9955	0,9987
	Test	0,6505	0,8338	0,8338	0,8375	0,8309	0,8306	0,8127	0,8133	0,8129	0,816	0,8155	0,8134	0,8144	0,814	0,814	0,8148
2	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9895	0,9905	0,9913	0,9917	0,9924	0,9929	0,9934	0,9938	0,9942	0,9946	0,9962
	Test	0,6505	0,8338	0,8338	0,8375	0,8309	0,8307	0,8137	0,8128	0,8134	0,8164	0,8155	0,8137	0,8126	0,8137	0,8147	0,8134
3	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9895	0,9905	0,9912	0,9917	0,9923	0,9927	0,9932	0,9935	0,9938	0,9942	0,9953
	Test	0,6505	0,8338	0,8338	0,8375	0,8309	0,8307	0,8141	0,8068	0,8075	0,8186	0,8097	0,8089	0,8165	0,8171	0,8096	0,8093
5	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9895	0,9904	0,9911	0,9915	0,992	0,9923	0,9926	0,9929	0,9931	0,9933	0,9936
	Test	0,6505	0,8338	0,8338	0,8375	0,8309	0,8307	0,831	0,8266	0,8328	0,8357	0,8312	0,8343	0,8298	0,83	0,8355	0,836
10	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9895	0,9903	0,9909	0,9912	0,9914	0,9916	0,9917	0,9918	0,9919	0,9919	0,9919
	Test	0,6505	0,8338	0,8338	0,8375	0,8309	0,8307	0,8309	0,8313	0,8314	0,8352	0,8346	0,8339	0,834	0,8342	0,8338	0,8338
15	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9895	0,9903	0,9907	0,9909	0,9911	0,9911	0,9912	0,9913	0,9913	0,9913	0,9913
	Test	0,6505	0,8338	0,8338	0,8375	0,831	0,8309	0,8314	0,8335	0,8333	0,8356	0,8331	0,8341	0,8348	0,8342	0,8346	0,8346
25	*Train*	0,8796	0,9846	0,9863	0,9878	0,989	0,9894	0,9901	0,9903	0,9904	0,9906	0,9906	0,9906	0,9907	0,9907	0,9907	0,9907
	Test	0,6505	0,8338	0,8338	0,8375	0,8305	0,833	0,8313	0,8328	0,8338	0,8364	0,8346	0,8351	0,8353	0,8353	0,8357	0,8357
40	*Train*	0,8796	0,9846	0,9863	0,9878	0,9888	0,9891	0,9897	0,9897	0,9899	0,99	0,99	0,99	0,99	0,99	0,99	0,99
	Test	0,6505	0,8338	0,8338	0,8375	0,8305	0,8325	0,8282	0,8311	0,8301	0,8308	0,8308	0,8308	0,8308	0,8308	0,8308	0,8308
50	*Train*	0,8796	0,9846	0,9863	0,9878	0,9888	0,9891	0,9896	0,9898	0,9898	0,9899	0,9899	0,9899	0,9899	0,9899	0,9899	0,9899
	Test	0,6505	0,8338	0,8338	0,8375	0,8305	0,8317	0,8275	0,8298	0,8298	0,8305	0,8305	0,8305	0,8305	0,8305	0,8305	0,8305

Tabulka 1: Výsledky přesnosti DecisionTree.

Zdroje dat:

GeeksforGeeks. *Decision Tree – GeeksforGeeks* (2025). Dostupné z: https://www.geeksforgeeks.org/machine-learning/decision-tree/

Wikipedia. *Decision tree* (2025). Dostupné z: https://en.wikipedia.org/wiki/Decision_tree

Atlassian. *Decision Tree: Definition & Examples* (2025). Dostupné z: https://www.atlassian.com/work-management/project-management/decision-tree

Turing. *Why Do We Use Decision Trees in Machine Learning?* (2022). Dostupné z: https://www.turing.com/kb/importance-of-decision-trees-in-machine-learning