Kristýna Janovská
janovkri

Podobnostní dotazování – Skyline

zpráva k projektu + pokyny ke spuštění aplikace

Popis projektu:

Cílem projektu bylo vytvoření aplikace umožňující doporučování produktů z dané databáze formou 2D skyline. Jako doporučované produkty jsem zvolila smartphony.
Vstupem, který volí uživatel, je dvojice atributů spolu s volbou, zda uživatel preferuje vyšší či nižší hodnoty daného atributu. Dále uživatel volí použitou metodu výpočtu, tedy jeden ze třech implementovaných algoritmů.
Výstupem jsou grafy či tabulka. Tabulka je interaktivní přehled položek databáze, které si uživatel může seřadit podle libovolného atributu. Grafy na základě použitých algoritmů a uživatelského vstupu zobrazují všechny položky v grafu s osami danými atributy. Položky ležící na skyline jsou spojeny čarou. Graf je interaktivní, při najetí myší na libovolný bod se uživateli zobrazí přehled všech atributů dané položky.

Způsob řešení:

Pro výpočet bodů ležících na skyline jsem použila tři algoritmy – plane-sweep, divide and conquer a přístup hrubou silou. Všechny tyto algoritmy jsem dále modifikovala, aby byly schopny počítat čtyři různé možnosti skyline podle preference vyšších či nižších hodnot.
Plane-sweep algoritmus:
Algoritmus pracuje s postupným přidáváním a ubíráním bodů (již seřazených podle jednoho z atributů) ze zásobníku v závislosti na velikosti druhého z atributů. Výsledkem algoritmu jsou body, které v zásobníku zůstanou, tyto body pak tvoří skyline.
Brute force algoritmus (přístup hrubou silou):
Porovnává se každý bod s každým, pokud je pro bod nalezen jiný bod, který má žádanější hodnotu obou atributů, méně žádaný bod neleží na skyline. Výsledkem je seznam bodů, které předčí ostatní body alespoň v jednom atributu.
Divide and conquer algoritmus:
Algoritmus, který rekurzivně rozděluje posuzované body seřazené podle jednoho z atributů na poloviny, na kterých je opět provedena rekurze. Dané poloviny se poté spojují dohromady a postupně formují skyline.

Implementace:

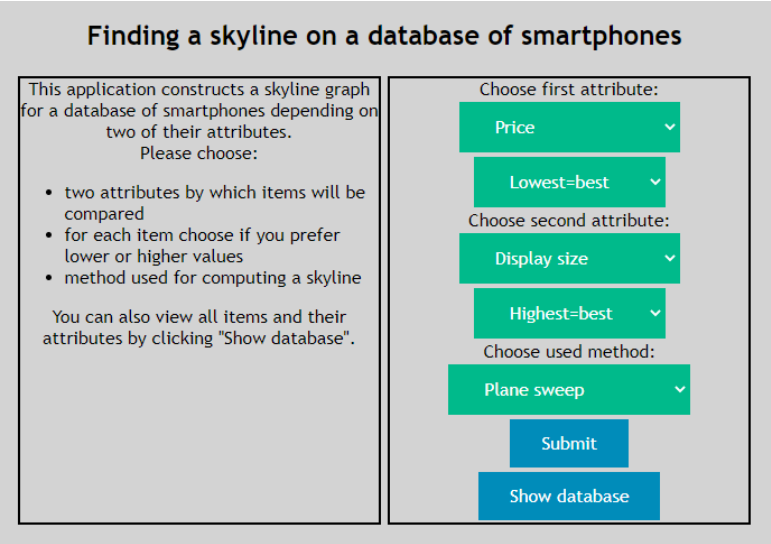
Práci jsem psala v jazyce Python za využití frameworků Flask a Dash. Flask jsem využila při předávání uživatelského vstupu programu, Dash zajistil následné vypsání interaktivního grafu a tabulky zobrazující databáze. Pro konstrukci grafu jsem využila knihovnu Pyplot. Pro práci s .csv souborem s daty jsem využila knihovnu Pandas. Z části je komunikace s databází zajištěna pomocí převodu na sqlite3 databázi.
Aplikace má několik základních částí:

- `app.py` – modul zajišťující webovou komunikaci a vykreslování dat za využití Flasku a Dashe
- `database.py` – modul zajišťující práci s databází pomocí třídy `Database`, její metoda `get_skyline(self, column1, column2, minmax, method)` přijímá uživatelský vstup na základě kterého využívá modulů pro jednotlivé algoritmy pro výpočet bodů skyline a na základě těchto výpočtů konstruuje graf
- `plane_sweep.py`, `divide_conquer.py`, `brute_force.py` – moduly pro jednotlivé algoritmy použité pro výpočet skyline
- `index.html` – úvodní stránka, na které si uživatel volí parametry skyline

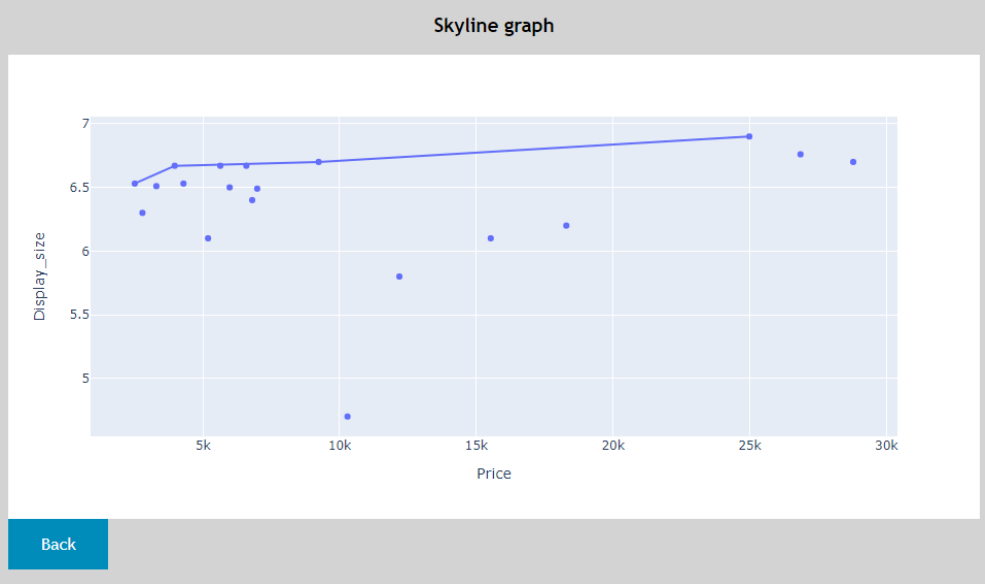
- `styles.css` – tyto soubory jsou dva, jeden pro část aplikace využívající Dash, druhý pro html úvodní stránku. Nastavují základní styly stránek.

Příklad vstupu a výstupu:

Vstup:



Výstup:



Experimentální sekce:

Přesnost všech algoritmů je stejná, mohou se však lišit v rychlosti. Jelikož časová složitost algoritmů plane-sweep a divide and conquer je O(nlogn), jsou tyto algoritmy rychlejší, než přístup hrubou silou, který má časovou složitost $O(n^2)$. V mé aplikaci se to však neprojevilo, protože jsem měla malý dataset.

Diskuze:

Protože jsem použila malý dataset, neprojeví se rozdíly v rychlosti algoritmů.
Některé z atributů, které si může uživatel zvolit, se nejeví jako vhodné pro konstrukci skyline, jelikož mají málo možných hodnot – například parametr počet jader telefonu dosahuje v datasetu pouze dvou možných hodnot.

Závěr:

Navrhla jsem aplikaci pro doporučování mobilních telefonů. Doporučování probíhá pomocí 2D skyline grafu, na němž jsou doporučené položky spojeny čarou. Uživatel volí, na základě kterých dvou atributů bude graf konstruován, jestli preferuje vyšší či nižší hodnoty těchto atributů a metodu použitou při výpočtu. Graf je interaktivní, uživatel při najetí myší na daný bod může zobrazit všechny atributy této položky. Metody pro výpočet skyline bodů jsem naimplementovala tři, plane-sweep algoritmus, algoritmus divide and conquer a přístup hrubou silou. Dále si uživatel může zobrazit tabulku zobrazující celou databázi. Tato tabulka je také interaktivní a uživatel může jednotlivé položky seřazovat podle hodnot jednotlivých atributů.

Spuštění aplikace:

Aplikace se spouští v terminálu příkazem:

- Python app.py (Windows)
- Python3 app.py (Linux)

Aplikace poté bude dostupná na adrese http://127.0.0.1:5000/

Pro spuštění aplikace je nutné mít nainstalované následující: Pandas, Pyplot, Dash, Flask.