Typy dolování webu: Content (obsah) / Structure (graf stránek a odkazy) / Usage (použití – logy)
WCM: crawling, identifikace textů, témat, NLP analýzy
WSM: analýza sociálních sítí, **WUM**: model chování uživatelů

Social web: uživatelé generují obrovské množství dat
Prosumer: uživatel je konzument a producent zároveň

Výzvy web structure mining: velikost webu, přetížení serverů, identifikace crawlerů – boti
Crawling: automatické získávání obsahu z daného zdroje
Bot: obsahuje frontu stránek k navštívení, plánovač, downloader a parser + databázi

Typy crawlerů: batch (jednorázově) / incremental (neustále obnovuje obsah) / focused (téma)
Seed page: první URL, kterou navštěvuji

Normalizace URL: z relativního odkazu vytvoříme absolutní, normalizujeme case, odstraníme schéma, nahrazení IP doménou, překódování znaků, odstranění query parametrů

Identifikace crawleru: hlavička User-Agent, dodržujeme etická pravidla v **/robots.txt**
robots.txt: User-agent, Disallow *(toto necrawluj)*, Sitemap, Host *(doména)*
Sitemap: strojově čitelný popis struktury webové stránky – XML se seznamem URL a jejich metadat
- loc *(adresa)*, lastmod *(datum změny)*, changefreq *(frekvence změn)*, priority

Revisit strategy: uniformní (pravidelně), proporciální (často měněné stránky) / hybridní
Freshness: poměr stránek, které se změnily

Ovládání indexování: noindex *(neukazovat)*, noarchive *(necachovat)*, none, nosnippet, notranslate

Při miningu dostáváme strukturovaná (schéma, čísla) / semistrukturovaná (CSV, XML) / nestrukturovaná (volný text) data => musíme z nich extrahovat relevantní informace.

Recogniser: část kódu, která získává z HTML kódu informace jako e-maily, adresy, telefonní čísla
Wrapper: pomůže převést semistrukturovaná data z webu na databázi
- manuální označení *(produkt, cena)*, indukcí *(najdi span.title)*, automatické *(matching techniky)*

Web Crawling: robot procházející a indexující weby vs **Scraping**: zaměřený na zisk dat z 1 stránky
Deep Web: obsah schován od crawling enginů *(za formuláři)* – DB uživatelů, e-mail schránky
- kontextuální *(závisí na poloze)*, dynamické stránky *(odpověď na dotaz)*, omezený přístup *(CAPTCHA)*
- soukromý web *(přihlášení)*, skriptovaný obsah *(AJAX)*, multimédia

Crawling dynamických stránek: Selenium – předvypočítání dotazů pro dané formuláře
Crawling omezených stránek: speciální agenti, využijí existující session / cookies pro přihlášení
Crawling AJAX aplikací: headless browsery, náročnější na výkon a zdroje

Spider Trap: množina webových stránek, které vytvoří nekonečné množství stránek pro crawling *(kalendář cal.org/01/2014)*
Honeypot Trap: stránky, které nejsou viditelné pro běžné uživatele, schovaná textová pole *(display: none)* – robot vyplní, uživatel ne

Machine Readable anotace: stránky ve strojově čitelném formátu – mikrodata, RDF-a, JSON-LD

Text mining: extrakce předem neznámých informací a dat z textu
- vysoká dimenzionalita problému, hodně mnohovýznamových slov, chybějící struktura

Předzpracování textu: převod do strojově rozpoznatelného formátu, identifikace slov (tokenů)
Tokenizace: rozdělení textu na slova *(bílé znaky jako oddělovače, čárka někdy to, někdy to)*

Frekvenční analýza: počítáme četnosti slov v textu, unikátní slova
- snížení dimenzionality, počtu tokenů a zvýšení jejich frekvence

Lematizace: nalezení základní formy slova *(jsem -> být)*
Stemming: nalezení kořene slova *(čekání -> čekat)*
Stop slova: nezajímavá slova *(spojky – a, the, it, they)*

Rozdělení na věty: identifikace vět v textu (tečka nemusí být konec věty – Mrs.)
N-gram: posloupnost více za sebou jdoucích slov (bigram, trigram – obsahuje informace o okolí slova)
Detekce frází: shlukování tokenů do jednotek *(named phrase – člen, přídavné + podstatné jméno)*

Detekce pojmenovaných entit (NER): něco, co má jméno *(osoba, lokace, místo)* – pomocí POS tagging
Part-of-speech tagging: přiřazení kategorií slovům v textu podle kontextu *(podstatné jméno, sloveso)*
Extrakce relací: hledáme sémantické relace mezi entitami *(osoba pracuje pro organizaci, účastní se …)*

opinion mining
Analýza sentimentu: pozitivní / negativní / neutrální – recenze, komentáře, politické názory
- na úrovni celého dokumentu nebo věty, zjistíme obecný názor lidí, detekce spamu, falešných názorů
- můžeme použít **opinion shifter** (not, but = opak), detekce sarkasmu (#sarcasm, /r)

Dictionary-based: každému slovu se přiřadí sentiment a spočítá se – nepozná sarkasmus, otázky
Učení sentimentu: supervizované *(anotovaná data – recenze produktů)*, nesupervizované *(fráze)*
Pointwise mutual informace: vzájemná informace o slovech – jedno je kladné, mělo by být i druhé

Rozšíření slovníku: manuálně, podle pozic slov a spojek *(beatiful and spacious)*, slovník **WordNet**
Specifické situace: akronymy, emotikony

Coreference resolution: dva různá pojmenování stejné entity – Mary Smith, Mrs. Smith

Sumarizace textu: snaha zredukovat text na jedno krátké shrnutí a abstrakt
- nalezení důležitého obsahu => seřazení => přeformulování, clean up

Bag of words: dokument a dotaz jsou množiny slov nebo termů
Boolský model: term je/není přítomný, Boolské operátory; **Vektorový model**: váha, tf-idf vážení

Term Frequency: kolikrát se term vyskytl v dokumentu (normalizované)
Inverse Document Frequency: běžnost termu ve všech dokumentech
Jaccardův koeficient: počítá překryv v množinách slov v jednotlivých dokumentech

Euklidovská vs **Kosinová** vzdálenost: dívá se na vzdálenost / úhel *(kosinová lepší)*
Word Embedding: na základě distribuce a kontextu slov se snaží vytvořit vektor v n-rozměrném prost.
CBOW: podle okolí vytváří vektor | **Skip-Gram**: podle vektoru vytváří okolí

FastText: rozšíření Word2Vec, rozděluje slovo podle slabik, umí pracovat i s novými slovy
Doc2Vec: embedování dokumentů do n-rozměrného prostoru

Latent Dirichlet allocation (LDA): dokument je popsán rozmístěním témat, téma rozmístěním slov
NLP modely: nadstavba nad word embedding, umožňují vnímat kontext (ELMo, BERT, GPT-3, GPT-4)

Analýza sociální sítě: analyzujeme grafy – osoby = uzly, vazby = hrany, můžou být ohodnocené
Typy analýzy: sociocentrická *(celá síť)*, egocentrická *(pohled jedince)*, znalostní *(který uzel rozpadne)*

Small world phenomenom: vzdálenosti mezi uzly jsou krátké
Six degrees of separation: mezi dvěma uzly je délka cesty maximálně 6
Dunbarovo číslo: každý člověk má průměrně 150 přátel

Hledání prostředního uzlu:
- podle **stupně** uzlu: najde uzly, které dokážou rychle rozšířit informaci
- podle **blízkosti**: průměr vzdáleností nejkratších cest *(kam všude dokážu z uzlu dosáhnout)*
- podle **betweeness**: kolik nejkratších cest prochází uzlem *(uzly na rozhraní komponent)*
- podle **vlastních vektorů**: uzel má vysokou hodnotu, pokud je napojen na další podobné uzly *(pomůže identifikovat uzly schované za dalšími)*

Velmi vzácné – častější lokální most: hrana, která významně snižuje vzdálenost mezi dvěma uzly
Most: hrana, která propojuje dva shluky mezi sebou *(jak se dostanu z jedné skupiny lidí do druhé)*
Embeddedness: počet společných sousedů dvou uzlů *(velký počet = větší důvěra)*

Triad closure: když A zná B, B zná C, pak by mělo taky A poznat C
Tranzitivita: propojování uzlů v silných skupinách (vytváříme kliky = úplné podgrafy)
+ slabé vazby mezi různými skupinami, propojeny mosty => **sociální síť**

Clustering coefficient: pravděpodobnost, že dva náhodní lidé v komunitě mají mezi sebou vazbu

Hustota grafu: poměr hran v grafu / hran v úplném grafu
Reciprocita (vzájemnost): kolik orientovaných hran A -> B platí i obráceně

Komponenta: skupina uzlů, které jsou spolu silně propojené
Singleton: uzel, který s nikým dále nekomunikuje

Power-Law: grafy typicky obsahují pár uzlů s vysokým stupněm
Homophily: princip – tíhneme k lidem, kteří mají podobné zájmy *(věk, práce, rasa)*
Motifs: hledání vzorů v grafu, rozlišuje různé sítě *(hvězda, řetěz, krabice, polo-klika, klika)*

Watts-Strogatz algoritmus: vytvoříme množinu n uzlů propojených do kruhu, propojíme nejbližší sousedy, náhodně přepojíme hrany s určitou pravděpodobností
Barabási-Albert algoritmus: začneme s jedním uzlem, postupně k němu připojujeme nové uzly
Erdos-Rényi: zadána pravděpodobnost vzniku hrany mezi 2 uzly *(1/n = větší, < 1/n malé komponenty)*

Komunita: množina uzlů, která má více vazeb uvnitř než vně
Kernighan-Lin algoritmus (hledání komunit): iterativně dělíme graf na 2 části, pro každý uzel vypočítám počet hran ve shluku a mimo něj a vyměním uzly, které mají nejlepší zlepšení
Girvan-Newman metoda: najdu hranu s největší betweenness, odstraním ji a přepočítám

Clique Percolation Method (hledání překrývajících se komunit): najdu všechny kliky dané velikosti k, vytvořím graf těchto klik, kliky jsou sousedící, pokud sdílí k – 1 uzlů

Predikce spojení: odhad pravděpodobnosti budoucího spojení
- podle **vzdálenosti**: největší pravděpodobnost je tam, kde je vzdálenost nejmenší
- podle **společných sousedů**: hrana vznikne tam, kde je nejvíce společných sousedů

Preferential attachement

Multimodální sítě: více než jeden typ uzlu *(lidé / zájmy / firmy)*, lze transformovat přes společné téma
Affliation network: svázání lidí pomocí organizací *(společných zájmů)* ; můžu pracovat s maticemi

Bow-tie struktura webu: SCC *(velká komponenta)*, IN: ukazuje do SCC, ale ne ven, OUT: dostaneme se tam ze SCC, ale nedostaneme se zpátky

PageRank: kvalita stránky určena kvalitou odkazů, prestiž přímo úměrná součtu prestiže stránek, které na ni odkazují, nezávisí na dotazu, odolná proti spamu, offline

Algoritmus: stránky začnou s popularitou 1/n, běží n iterací, spočte se, převáží počtem stránek
Problémy: stránky, které nemají žádný inlink, cyklické odkazy
=> řešení: **náhodný teleport** *(damping factor – 85 % respektuji graf, 15 % náhodný teleport)*

ireducibilní = dostanu se z každého uzlu všude (silně souvislý) – v matici nejsou nuly

HITS: vypočítáván až v okamžiku dotazu, hledá nejlepší **huby** *(odchozí linky)* a **autority** *(příchozí linky)*
Algoritmus: najdu čistě top k položek, které odpovídají dotazu => udělám jejich graf, napočítám hub/authority skóre, v praxi se příliš neuchytil

Graph Embeddings: reprezentace velkého grafu ve vektorech menší dimenze tak, aby zachoval důležité informace *(embedding komponent, podgrafů, grafu)* => Node2Vec

Web usage mining: analýza chování uživatelů, provázání obsahu strukturou, přizpůsobení webu
- části: předzpracování, hledání vzorů, analýza vzorů => optimalizace, personalizace, marketing, RS

Data o používání: primární zdroj, IP adresa, čas přístupu, zdroj, parametry, cookies – page views
Data o obsahu a struktuře: textová data, multimédia, sémantika *(historie odkazů)* i struktura *(obsah)*
Data o uživateli: registrační formuláře, hodnocení uživatelů, zájmy (explicitní) / implicitní zájmy

Implicitní sběr informací: web / search logy, browser / desktop agenti, CSS / **Javascript trackery**
JS trackery: randomizovaný identifikátor uživatele, čas přístupu, referer, user-agent, přenos v cookies

Předzpracování: vyčištění neúspěšných / nezajímavých požadavků *(obrázky)*, identifikace uživatele *(cookies, IP, user-agent)* a session *(sekvence návštěvy stránek, podle času)* => reprezentujeme v user-pageview nebo transaction matrix

Integrace sémantické informace: obohacení o produkt, typ, informace – pomocí násobení matic
Analýza asociací: hledání skupin stránek, které se společně navštěvují *(slevy > produkt > košík)*

Apriori algoritmus: 1. vygeneruje časté množiny *(se supportem větším než min_support)* a z nich 2. vygeneruje pravidla *(podle minimální confidence)*
Asociační pravidlo: implikace X -> Y *(antecedent -> consequent)*
Support: kolikrát si lidi koupí X (X / all) **Confidence**: kolik lidí, co si koupilo X, si koupí i Y (X&Y / X)

=> Neřeší časovou návaznost pravidel *(k tomu Markovské řetězce)*.

Web Analytics: analýza návštěvnosti *(e-commerce: conversion, revenue, impression)*
- zajímají nás unikátní zobrazení stránek, počet návštěv, prům. délka návštěvy, vracející se uživatelé
Čas strávený na stránce: pozor na přepínání záložek, Javascript *onbeforeunload / visibilitychange*

Conversion: kolikrát člověk něco koupil / počet návštěv
Mikrokonverze: ohodnocení produktu, zhlédnutí videa **Makrokonverze**: odeslání objednávky

Bounce rate: kolik lidí navštívilo na celém webu jen **jednu** stránku *(na e-shopu nežádoucí)*
Conversion funnel: trychtýř konverzí, sledujeme dílčí kroky zákazníků *(100 % homepage -> 60 % stránka produktu -> 30 % umístí do košíku => 3 % něco reálně koupí)*
Landing page: vstupní stránka **Exit page**: poslední stránka daného session

Doporučovací systémy (RS): řeší problém přehlcení informacemi, hodně možností
- vstup: minulé chování (logy, analytiky), explicitní feedback, demografická data
- klíčová myšlenka **personalizace** (přizpůsobit se individuálním potřebám jednotlivce)

Základní vstupy RS: personalizace *(uživatelův profil a kontext)*, kolaborativní filtrování *(data ostatních uživatelů)*, content-based *(vlastnosti produktů)*, knowledge-based *(znalostní modely)*

Kolaborativní filtrování: nezajímá nás obsah doporučovaných předmětů, ale **podobnost** mezi preferencemi jednotlivých uživatelů
- **user-based**: uživatelé mně podobní - **item-based**: předměty podobné těm, které jsem likenul

User-based doporučování: zjistím hodnocení položek, které jsem ještě nehodnotil, ale ostatní mně podobní sousedů už ano => musíme mít aktivního uživatele *(doporučím nejpopulárnější)*, problém prvního hodnotícího *(doporučuji i náhodně)*
Item-based doporučování: dívám se podle hodnocení itemů a ne uživatelů

Kosinová podobnost: kosinus úhlu mezi vektory A a B dělený jejich normami
Pearsonova podobnost: korelace obou dvou vektorů *(pozitivní i negativní)*

Maticová faktorizace: velkou matici zkouším rozložit na více matic, hledám skryté (**latent**) faktory
=> uživatele zajímá pouze sci-fi, můžu se zaměřit pouze na sci-fi

Content-based filtrování: sleduji charakteristiky obsahu *(kategorie, anotace, data z text miningu)*
- reprezentuji obsah vektorem, reprezentuji uživatele vektorem kategorií / anotací
- nepotřebuji množinu hodnotících uživatelů, poskytuje **důvod** doporučení *(máš rád sci-fi)*
- problém s uživateli bez profilu, neřeší interakce ostatních uživatelů

Serendipity: překvapivost *(jednou jsem koupil lednici => bude mi ji to doporučovat pořád?)*

Znalostní RS: řeší situace, kde se nedají vytvořit feature vektory nebo hodnotit *(nemovitosti, pojištění)*
- constraint-based *(uživatel zadá omezení = dům s 3 pokoji, do 5 mil. Kč)*
- case-based *(chceš něco takového, uživateli?)*

Kontextové RS: čas, poloha, sociální informace *(vánoční stromek)* **Hybridní RS**: vážení dílčích RS

Evaluace RS: off-line *(na části dat se naučí, na druhém zkontroluje metriky)*, user studies *(testovací skupina, ta ohodnotí)*, on-line *(A/B testování)*
Metriky: precision *(rel|ret)*, recall *(ret|rel)*, F-measure, serendipity, accuracy *(přesnost)*
chtěl vědět přesný vzorec pro všechny metriky (i accuracy, F-measure)
Útoky na RS: push *(vylepšit můj produkt)* / nuke *(poškodit produkty ostatních)*
- **random**: generování náhodných profilů - **average**: hodnocení položek průměrem
- **bandwagon**: kladně hodnotím populární, náhodně ostatní - **reverse** bandwagon: negativně nepop.
- **segment**: podle zájmu části uživatelů - **love** / **hate**: svému produktu max/min, ostatním min/max
- **probe**: učí se na základě falešného profilu

Řešení útoků: ochrana před roboty pomocí CAPTCHA, detekční algoritmy *(supervised -> detekce známých útoků, unsupervised -> hledá příliš podobné profily, příliš zaujaté)*

Data streams: data rostou napříč časem, generují se nová, hodnotu musíme ukládat okamžitě (data se nevejdou do operační paměti / jsou průběžně generována v čase) => data ze senzorů, sociálních sítí
Vlastnosti data streams: velké množství dat (volume), velká rychlost (velocity), mění se za běhu

Streamovací přístup: uděláme náčrt (**sketch**) na zjednodušeném vzorku dat
Axiomy: položka prochází jen jednou, na položku je málo času a místa, stream se mění v čase, odpovědi musí být v reálném čase => přibližné odpovědi jsou v pořádku, někdy potřebujeme náhodu

Vzorkování: hledáme rovnoměrné náhodné vzorky v zadaném proudu (streamu) dat

Reservoir sampling: prvních k záznamů přijde bez problému, další prvky vkládám s pravděpodobností p a pokud mám vložit, odeberu náhodný prvek => stream může obsahovat duplicity plýtvající místem
Moving window: dívám se na okénko k posledních dat *(řeší expiraci starých dat)*

Sketches: snaží se nasamplovat data malým vzorkem, aby reprezentoval celá data *(distribuční funkce)*

Četnosti prvků: naivně n čítačů pro každou položku, složitější: pamatuji si pravděpodobnost pro každou položku a ukládám každou p-tou položku

Count-Min Sketch: hashovací funkce s omezeným počtem čítačů, aproximuji hodnoty pomocí w sloupců a d řádků, každý řádek má hash funkci a počítám minimum hodnot v daném řádku

Lossy Counting: tvořím si histogram okna, na konci okénka snížím výpočet o 1

Hledání častých množin: načtu co nejvíce hodnot, napočítám vzory a provedu Lossy Counting

Flajolet-Martin: hledá počet různých prvků – zahashuji data, pro každý prvek spočítám počet nul na konci řetězce a držím si maximální počet nul => unikátních položek by mělo být cca 2^n

Technologie pro data stream mining: Apache Storm, Samza, Apache Spark, Flink

ALGORITMY:

Generování sítě: Watts-Strogatz, Barabási-Albert, Erdos-Rényi
Hledání komunit: Kernighan-Lin, Girvan-Newman
Překryv komunit: Clique Percolation Method

Při vyhledávání stránek: PageRank + Google matice, HITS
Asociační pravidla: Apriori algoritmus

Doporučovací systémy: Kolaborativní filtrování (user / item based), Content-based filtrování

Data streams: Reservoir sampling, Moving window
Počítání četnosti prvků: Count-Min Sketch, Lossy Counting
Počet různých prvků: Flajolet-Martin