9. Základní pojmy z datové analýzy

9.1 CO JE TO DATOVÁ ANALÝZA?

9.1.1 DATA JSOU VŠUDE

Data jsou generovány nevídanou rychlostí stroji, lidmi a věcmi. Předpovídá se globální růst internetového provozu a trendu širokopásmového připojení pro mobilní a pevné sítě. Množství datového provozu i množství IP připojených zařízení se každý rok několikrát násobí. Video tvořilo v roce 2020 79 % internetového provozu.

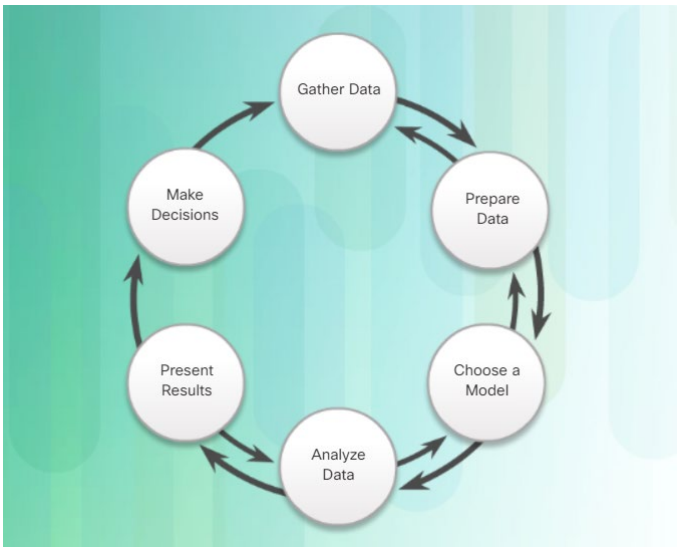
Data jsou generována ze senzorů, zařízení, zvuku, videa, sítí, log souborů, transakčních aplikací, webu a sociálních médií. Velký objem a rozmanitost datových souborů je klíčovou vlastností, která odlišuje data od velkých dat.

Vznik velkých datových souborů vyžaduje pokročilé metody, technologie a infrastruktury pro jejich zpracování a převedení na **smysluplné informace**. Data již nelze ukládat na několika strojích ani je nelze zpracovávat jedním strojem.

Společnosti aktivně vytvářejí profily a zpracovávají data o svých systémech, uživatelích a procesech, aby podnítily růst a inovace. Výzkumníci a analytici hledají způsoby, jak získat přístup a analyzovat data, která byla kdysi považována za nepoužitelná. Pokročilé analytické techniky lze použít u velkých souborů dat, jako je **textová analýza**, **strojové učení**, **prediktivní analytika**, **dolování dat**, **statistika a zpracování přirozeného jazyka**. Podniky mohou analyzovat dříve nevyužité zdroje dat nezávisle nebo společně se svými stávajícími podnikovými daty a získat tak nové poznatky. Tyto poznatky vedou k výrazně lepším a rychlejším rozhodnutím.

9.1.2 CRISP-DM

Existuje mnoho metodologií pro provádění analýzy dat, včetně populárního modelu **Cross Industry Standard Process for Data Mining** (CRISP-DM).



Obr. Životní cyklus analýzy dat

Životní cyklus analýzy dat, který se velmi podobá vědecké metodě, je navržen pro použití v obchodním prostředí. Všimněte si, že mezi některými kroky směřují šipky oběma směry. To podtrhuje skutečnost, že životní cyklus může vyžadovat mnoho opakování, než si osoby s rozhodovací pravomocí budou dostatečně jistí, aby se posunuli vpřed.

9.1.3 MOŽNOSTI NÁSTROJŮ PRO ANALÝZU DAT

Dříve se při omezeném množstvím dat využíval pečlivý procese ruční analýzy. Dnes, s masivním růstem množství dat, jsou využívány počítače a příslušný software, aby získaly přehled o obchodních vzorech a daly všem těmto datům smysl.

Nástroj, který se má použít, závisí na typu analýzy, která má být provedena. Některé nástroje jsou navrženy tak, aby zvládly manipulaci a vizualizaci velkých souborů dat. Ostatní nástroje jsou navrženy s komplexními matematickými modelovacími a simulačními schopnostmi pro predikci a prognózování.

Bez ohledu na použité nástroje by měly být schopny splňovat tyto funkce:

- **Snadné použití** – Nástroj, který se snadno učí a používá, je často efektivnější než nástroj, který se obtížně používá. Také nástroj, který se snadno používá, vyžaduje méně školení a méně podpory.
- **Manipulace s daty** – Software by měl uživatelům umožnit čistit a upravovat data, aby byla lépe použitelná. To vede k tomu, že data jsou spolehlivější, protože anomálie lze detekovat, upravit nebo odstranit.
- **Sdílení** – každý musí sledovat stejné soubory dat, aby mohl efektivně spolupracovat. To pomáhá lidem interpretovat data stejným způsobem.
- **Interaktivní vizualizace** – Chcete-li plně porozumět tomu, jak se data mění v průběhu času, je důležité vizualizovat trendy. Základní tabulky a grafy nemohou plně znázornit, jak se informace vyvíjejí tak, jak se může vyvíjet tepelná mapa nebo zobrazení časového pohybu.

9.1.4 ROLE PYTHONU V ANALÝZE DAT

Python byl vytvořen v roce 1991 jako snadno naučitelný jazyk s mnoha knihovnami používanými pro manipulaci s daty, strojové učení a vizualizaci dat. Díky použití těchto knihoven se programátoři nemusí učit více programovacích jazyků nebo trávit čas učením se, jak používat mnoho různých programů k provádění funkcí těchto knihoven.

Jupyter Notebook umožňuje existenci instrukcí a programování ve stejném souboru. Je snadné měnit kód v poznámkových blocích a experimentovat s tím, jak lze použít různé kódy k manipulaci, analýze a vizualizaci vašich dat.

Toto jsou některé z knihoven jazyka Python:

- **NumPy** – Tato knihovna přidává podporu pro pole a matice. Má také mnoho vestavěných matematických funkcí pro použití v datových souborech.
- **Pandas** – Tato knihovna přidává podporu pro tabulky a časové řady. Pandas se mimo jiné používá k manipulaci s daty a jejich čištění.
- **Matplotlib** – Tato knihovna přidává podporu pro vizualizaci dat. Matplotlib je vykreslovací knihovna schopná vytvářet jednoduché čárové grafy až po komplikované 3D a obrysové grafy.

9.2 VYUŽITÍ BIG DAT

9.2.1 PROČ ANALYZOVAT BIG DATA

9.2.1.1 Big Data a rozhodování

Škálovatelné technologie, které umožňují distribuované výpočty a virtualizace, umožňují správcům datových center spravovat tři nejdůležitější ze čtyř aspektů velkých dat:

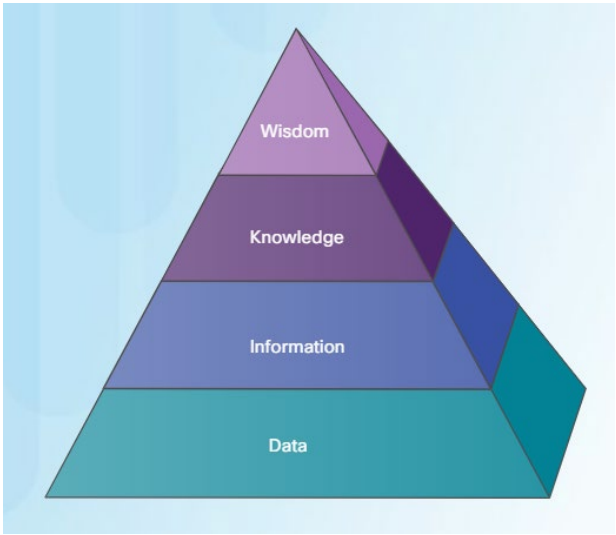
- **Objem** (Volume)
- **Rychlost** (Velocity)
- **Rozmanitost** (Variety)

Statistické metodiky zabudované v aplikacích umožňují datovým analytikům interpretovat a používat Big Data k lepším rozhodnutím. Moderní nástroje pro analýzu dat umožňují extrahovat a transformovat nezpracovaná data za účelem vytvoření mnohem menšího souboru kvalitních dat. **Samotná data však nejsou smysluplnou informací, data je nutné analyzovat a následně prezentovat ve formě, kterou lze dále srozumitelně interpretovat.** To je to, co tvůrci rozhodnutí potřebují, aby přijali správná opatření.

9.2.1.2 Data, Informace, Znalost, a Moudrost

Níže je uveden příklad každé úrovně pyramidy zdola nahoru:

- **Data** – Shromažďujte údaje o teplotě z více geolokalizovaných senzorů.
- **Informace** – Extrahujte časové a lokalizační údaje. Výsledek ukazuje, že teploty neustále globálně rostou.
- **Znalosti** – Ověřte několik hypotéz a vyjde najevo, že nárůst je zřejmě způsoben lidskou činností, včetně emisí skleníkových plynů.
- **Moudrost** – Pracujte na snížení emisí skleníkových plynů.



Obr. Pyramida DIKW

9.2.2 TYPY ANALÝZY DAT

9.2.2.1 Deskriptivní analýza

Výběr typu datové analýzy, který je třeba implementovat, bude záviset na problému, který je třeba vyřešit, nebo na otázkách, které je třeba zodpovědět.

Tento kurz probere tři typy datové analýzy:

- Deskriptivní
- Prediktivní
- Preskriptivní

Deskriptivní analytika primárně využívá pozorovaná data. Používá se **k identifikaci klíčových charakteristik souboru dat**. *Souhrnná data z deskriptivní analýzy poskytují informace o předchozích událostech a trendech ve výkonu*. *Deskriptivní analytika se spoléhá pouze na historická data*, aby poskytovala pravidelné zprávy o událostech, které se již staly. Tento typ analýzy se také používá ke generování ad hoc zpráv, které shrnují velké množství dat k zodpovězení jednoduchých otázek jako **„kolik...“** nebo **„co se stalo“**. Lze jej také použít k prohloubení dat a položení hlubších otázek ke konkrétnímu problému. *Cílem deskriptivní analýzy je shrnout vaše data do kompaktnějších a užitečnějších informací*.

Příkladem deskriptivní analýzy je hodinová zpráva o provozu.

9.2.2.2 Prediktivní analýza

Prediktivní analytika se na základě dat a statistik pokouší s jistou mírou spolehlivosti **předvídat, co se může stát dál**. Prediktivní analytiku lze použít k *odvození chybějících dat* a *vytvoření budoucí trendové linie na základě minulých dat*. Využívá simulační modely a předpovědi, aby navrhl, co by se mohlo stát.

Příkladem prediktivní analýzy je počítačový model, který využívá Big Data k **předpovědi počasí**.

Dalším způsobem, jak se podívat na prediktivní analýzu, je vytváření nových dat tak, že začneme se stávajícími daty. Představte si, že chcete prodat svůj dům a nevíte, jakou cenu za něj stanovit. Jako údaj o ceně můžete vzít ceny nedávných prodejů domů v sousedství a vlastnosti těchto domů (např. počet ložnic, koupelna, stav atd.). Váš dům ale pravděpodobně není totožný s žádným z ostatních domů. Zde může pomoci prediktivní analytika. Prediktivní model pro cenu je založen na údajích, které máte o předchozích prodejích. „Předpovídá“ vhodnou cenu za váš dům.

Dalším příkladem je klasifikace. Pokud například dostanete tweet nebo příspěvek, klasifikujte tweet jako pozitivní nebo negativní na základě textu, který obsahuje.

9.2.2.3 Preskriptivní analýza

Preskriptivní analytika **předpovídá výsledky a navrhuje kroky, které budou mít největší přínos** pro podnik nebo organizaci. Preskriptivní analytika doporučuje akce nebo rozhodnutí na základě komplexní sady cílů, omezení a voleb. Může být použit k navrhování, jak zmírnit nebo dokonce předejít rizikům. Preskriptivní analytické implementace mohou vyžadovat systém zpětné vazby pro sledování výsledku přijatých akcí. Příkladem preskriptivní analýzy je počítačový model, který využívá velká data k vytváření doporučení akciového trhu k nákupu nebo prodeji akcií.

Pro analýzu dat se používají všechny tři typy analýz.

Prescriptive Analytics

Type	Tasks	Questions
Descriptive	Standard Reporting	What happened?
	Ad Hoc Reporting	How many, how often, where?
	Data Queries	What exactly is the problem?
Predictive	Simulation	What could happen?
	Forecasting	What if these trends continue?
	Predictive Modeling	What will happen next?
Prescriptive	Optimization	How can we have the best outcome?
	Optimizations Under Uncertainty	How can we have the best outcome, given variability?

9.2.3 VČASNÁ ANALÝZA VELKÝCH DAT

9.2.3.1 Role času v analýze dat

Před érou velkých dat byla role času v analýze dat omezena na to, jak dlouho trvalo sestavení souboru dat z různých zdrojů, nebo jak dlouho trvalo provedení souboru dat pomocí nějakého výpočtu.

Data neustále generují senzory, spotřebitelé, uživatelé sociálních sítí, tryskové motory, burza a téměř cokoli jiného, co je připojeno k síti. Tato *data nerostou jen co do kvantity, mění se také v reálném čase*. *Analýza dat musí být také prováděna v reálném čase během sběru dat*.

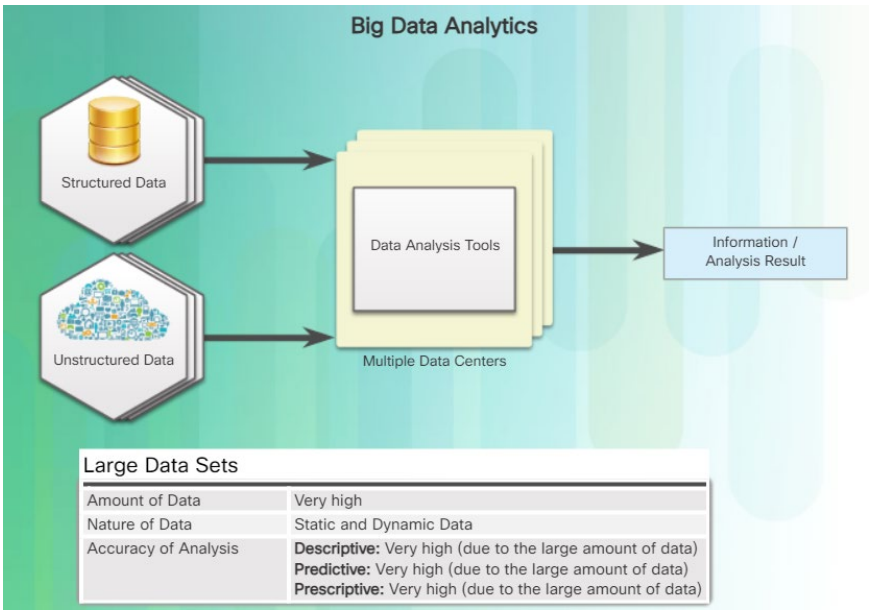
Rozhodnutí založená na datech mohou mít následující výhody:

- Delší čas na výzkum a vývoj produktů a služeb
- Zvýšená efektivita a rychlejší výroba
- Rychlejší doba uvedení na trh
- Účinnější marketing a reklama

9.2.3.2 Od tradiční analýzy k analýze velkých dat

V minulosti, kdy byla většina datových souborů relativně malá a dobře spravovatelná, mohli analytici používat tradiční nástroje (Excel, nebo statistický program) k vytváření smysluplných informací z dat. Soubor dat obvykle obsahoval historická data a zpracování těchto dat nebylo vždy časově závislé. Tradiční databáze musely být navrženy předtím, než mohla být data zadána. Poté by bylo možné data, pokud nejsou příliš velká, čistit, filtrovat, zpracovávat, sumarizovat a vizualizovat pomocí tabulek, grafů a dashboardů.

S rostoucím objemem, rychlostí a rozmanitostí datových sad se složitost ukládání, zpracování a agregace dat stává výzvou pro tradiční analytické nástroje. Velké datové sady lze distribuovat a zpracovávat na více geograficky rozptýlených fyzických zařízeních i v cloudu. Pro tyto velké soubory dat jsou potřeba nástroje pro velká data, jako jsou *Hadoop* a *Apache Spark*, aby umožnily analýzu v reálném čase a prediktivní modelování.



Obr: Analýza velkých dat

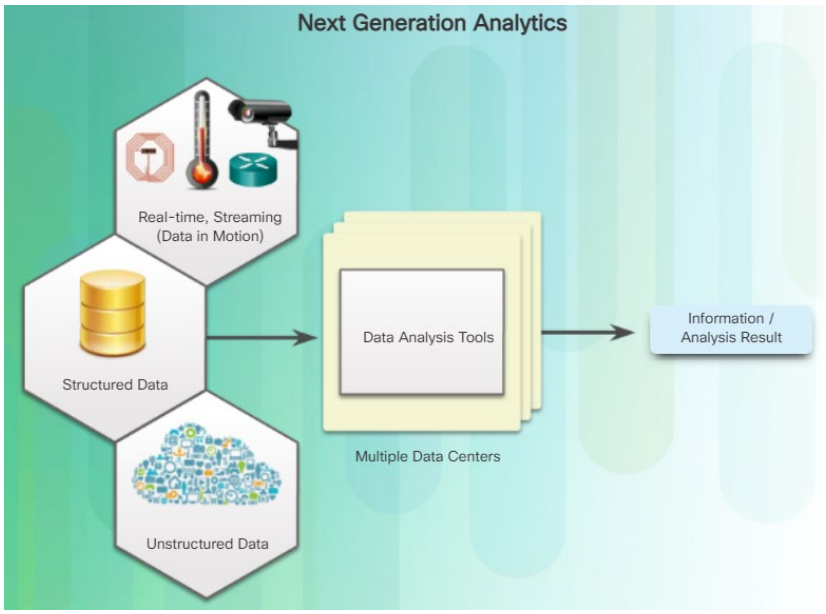
9.2.3.3 Analytika nové generace

Aby podniky mohly činit optimální rozhodnutí, již nestačí shromažďovat data z předchozího fiskálního roku a spouštět typy dotazů s popisnou analýzou. Stále více je nutné používat nástroje prediktivní a preskriptivní analýzy, abychom zůstali konkurenceschopní ve světě, ve kterém se tempo změn zrychluje. Analytika nové generace se nemusí spoléhat pouze na provádění statistických analýz na celém souboru dat, jak tomu bylo u tradičních analytických nástrojů. Vzhledem k obrovskému množství datových bodů a atributů shromážděných o záznamu, nebo věci, lze z pokročilých analýz získat nové chování a poznatky, které zlepšují předpovědi a přesnost předpovědi.

Chcete-li například provést úpravy rozhodnutí v reálném čase, lze odpovědět na následující otázky:

- Které akcie budou mít s největší pravděpodobností nejvyšší denní zisk na základě obchodování za poslední hodinu?
- Jaký je nejlepší způsob, jak dnes odpoledne nasměrovat nákladní vozy na základě ranních prodejů, stávajících zásob a aktuálních dopravních zpráv?
- Jaká údržba je vyžadována pro letoun na základě údajů o jeho výkonu generovaných během posledního letu?

Manipulace s těmito strojově generovanými daty v kombinaci s geografickými daty, počtem zařízení generujících data, rozmanitostí výrobců zařízení, frekvencí generování dat a celkovým objemem dat vyžaduje nový infrastrukturní software. Tento infrastrukturní software musí být schopen distribuovat výpočty a ukládání dat mezi edge, fog a cloud tam, kde to lépe vyhovuje potřebám podniku.

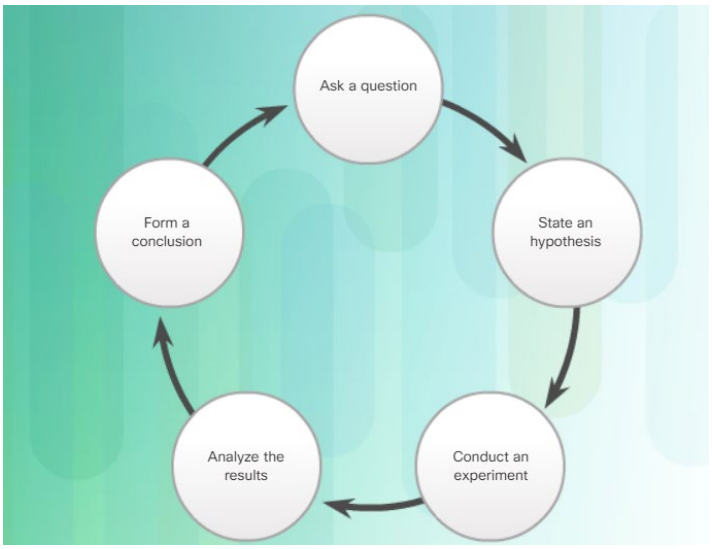


Obr. Analytika nové generace

9.2.4 ŽIVOTNÍ CYKLUS DATOVÉ ANALÝZY

9.2.4.1 Vědecká metoda

Proces, který datový analytik používá k vytváření závěrů, je velmi podobný vědecké metodě znázorněné na obrázku. Datový analytik si může položit otázku: „V *jaké čtvrti v San Franciscu bylo mezi 1. červnem a 1. srpnem 2014 nejvíce hlášených zločinů?*“ Vědec možná bude chtít vyřešit problém: „*Proč krev mladé myši zvrátí účinky stárnutí, když ji vložíme do starší myši*? Bez ohledu na použitou metodu nebo postup dokončí jak datoví analytici, tak vědci proces, který zahrnuje kladení otázek, sběr dat, analýzu dat a vyvozování závěrů nebo prezentaci výsledků.



Obr. Životní cyklus analýzy dat: vědecká metoda

9.2.4.2 Obchodní hodnota

Analýza dat umožňuje podnikům lépe porozumět dopadu jejich produktů a služeb, upravit své metody a cíle a rychleji poskytovat svým zákazníkům lepší produkty. Schopnost získat nové poznatky ze svých dat přináší obchodní hodnotu.

Michel Porter z Harvardu popisuje, jak IT potřetí za 50 let přetvořilo byznys:

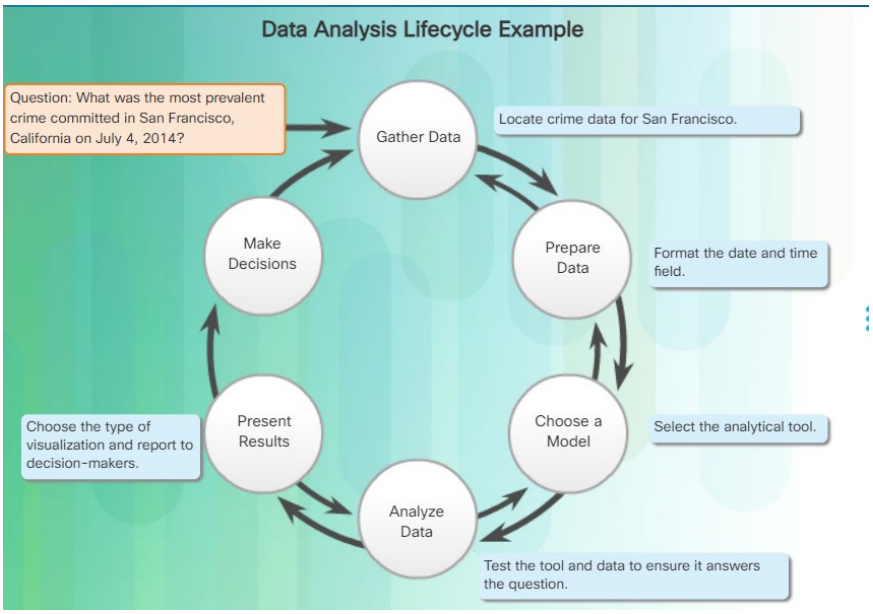
***První vlna IT**, během 60. a 70. let, automatizovala dílčí činnosti, jako je vyplácení zaměstnaneckých mezd, nebo podpora návrhu a výroby produktů. **Druhou vlnou** transformace obchodu byl vzestup internetu v 80. a 90. letech 20. století, který umožnil koordinaci a integraci externích dodavatelů, distribučních kanálů a zákazníků napříč geografickými oblastmi.*

*S IoT jsme nyní ve **třetí vlně, IT** se stává nedílnou součástí samotného produktu. Vestavěné senzory, procesory, software a konektivita v produktech (ve skutečnosti jsou počítače umístěny do produktů) spolu s cloudem, kde se ukládají a analyzují produktová data a spouštějí se některé aplikace, vedou k dramatickým vylepšením funkčnosti a výkonu produktu. Obrovské množství dat získaných používáním nových produktů umožňuje mnohá vylepšení.*

9.2.4.3 Příklad životního cyklu analýzy dat

Stejně jako u vědecké metody začíná životní cyklus analýzy dat otázkou. Mohli bychom si například položit otázku: „*Jaký byl nejčastější zločin spáchaný v San Franciscu v Kalifornii 4. července 2014*?“ Každý krok životního cyklu analýzy dat zahrnuje mnoho úkolů, které musí být dokončeny před tím, než se přejde k dalšímu kroku. Na obrázku je zobrazen pouze jeden příklad úlohy. Následuje stručný popis každého kroku:

- **Shromažďování dat** – Proces vyhledání dat a následné rozhodnutí, zda je k dispozici dostatek dat k dokončení analýzy. *V tomto případě bychom hledali otevřený datový soubor statistik kriminality pro San Francisco během července 2014.*
- **Příprava dat** – Tento krok může zahrnovat mnoho úkolů k transformaci dat do formátu vhodného pro nástroj, který bude použit. *Soubor údajů o trestné činnosti již může být připraven k analýze*. Obvykle je však třeba provést určité úpravy, které pomohou odpovědět na naši otázku.
- **Výběr modelu** – Tento krok zahrnuje výběr techniky analýzy, která nejlépe odpoví na otázku pomocí dostupných dat. Po výběru modelu je vybrán nástroj (nebo nástroje) pro analýzu dat. V této kapitole se naučíte používat jazyk Python a knihovny Pythonu k přípravě, analýze a prezentaci dat.
- **Analýza dat** – Proces testování modelu postavíme proti datům a rozhodneme, zda jsou model a analyzovaná data spolehlivé. *Dokázali jste odpovědět na otázku pomocí vybraného nástroje?*
- **Prezentace výsledků** – Toto je obvykle poslední krok pro datové analytiky. Je to proces sdělování výsledků těm, kdo dělají rozhodnutí. Někdy je datový analytik požádán, aby doporučil akce. *Pro údaje o kriminalitě ze 4. července by bylo možné použít sloupcový graf, koláčový graf nebo jiné znázornění ke sdělení toho, která kriminalita byla nejrozšířenější. Analytik by mohl navrhnout zvýšení přítomnosti policie v určitých oblastech, aby se zabránilo zločinu v konkrétní svátek, jako je 4. července*.
- **Rozhodování** – poslední krok v životním cyklu analýzy dat. Organizační vedoucí začleňují nové znalosti jako součást celkové strategie. Proces začíná znovu shromažďováním dat.



Obr. Životní cyklus analýzy dat: příklad

9.2.5 ZDROJE DAT

9.2.5.1 Soubory

Existuje mnoho různých zdrojů dat. Obrovské množství historických dat lze nalézt v souborech, jako jsou dokumenty **MS Word**, **e-maily**, **tabulky**, **MS PowerPoint**, **PDF**, **HTML** a soubory ve formátu **prostého textu**. Toto je jen několik typů souborů, které obsahují data.

Velká data lze také nalézt ve veřejných a soukromých archivech. Naskenované papírové archivy obsahující historická data z různých zdrojů jsou to určitě Big Data. Například ve formulářích a fakturách, zdravotních pojištění, daňových přiznání a interakcí se zákazníky a daňových dokladech je obrovské množství údajů. Tento seznam je jen malou částí archivovaných dat.

Uvnitř organizací jsou nezpracovaná data vytvářena prostřednictvím systému *řízení vztahů se zákazníky*, systému *řízení výuky*, systému a *záznamu lidských zdrojů*, *intranetu* a dalších procesů.

Různé aplikace vytvářejí soubory v různých formátech, které nemusí být nutně vzájemně kompatibilní. Z tohoto důvodu je zapotřebí univerzální formát souboru. Soubory **CSV** (*comma-separated values*) jsou typem souboru ve formátu prostého textu popsaného v RFC 4180. *Soubory CSV používají čárky k oddělení sloupců v tabulce dat a znak nového řádku k oddělení řádků*. Každý řádek je záznam. Ačkoli se běžně používají pro import a export v tradičních databázích a tabulkových procesorech, neexistuje žádný konkrétní standard.

JSON a **XML** jsou také typy souborů ve formátu prostého textu, které používají standardní způsob reprezentace datových záznamů. Tyto formáty souborů jsou kompatibilní s celou řadou aplikací. Převod dat do společného formátu je cenným způsobem, jak kombinovat data z různých zdrojů.

9.2.5.2 Internet

Internet je dobré místo pro vyhledávání velkých dat. Najdete zde *obrázky*, *videa* a *audio*. Veřejná *webová fóra* také vytvářejí data. *Sociální média,* jako je YouTube, Facebook, instant messaging, RSS a Twitter, to vše se řadí ke zdrojům dat na internetu. Většina těchto dat je **nestrukturovaná**, což znamená, že není snadné je kategorizovat do databáze bez určitého typu zpracování.

Webové stránky jsou vytvořeny, aby poskytovaly data lidem, nikoli strojům. Nástroje „**Web scraping**“ automaticky extrahují data ze stránek HTML. Je to podobné jako **Web Crawler** nebo **Spider** vyhledávače. Prozkoumává web, aby extrahovala data a vytvořila databázi, která bude reagovat na vyhledávací dotazy.

Web Scraping software může používat Hypertext Transfer Protocol nebo webový prohlížeč pro přístup na World Wide Web. Web scraping je obvykle automatizovaný proces, který využívá Bota, nebo Web Crawler. Konkrétní *data se shromažďují a kopírují z webu do databáze nebo tabulky*. *Data pak lze snadno analyzovat*.

Pro implementaci web scrapingu musí proces nejprve stáhnout webovou stránku a poté z ní extrahovat požadovaná data. Web scraper se může například používat k vyhledávání a kopírování jmen, telefonních čísel a adres.

Kromě dolování kontaktů se web scraping používá pro dolování dalších typů dat, jako jsou výpisy nemovitostí, data o počasí, průzkumy a srovnávání cen. Mnoho velkých poskytovatelů webových služeb, jako je Facebook, poskytuje standardizovaná rozhraní pro automatický sběr dat pomocí rozhraní API. Nejběžnějším přístupem je použití **RESTful** *aplikačního programového rozhraní (API)*. *RESTful API používají HTTP jako komunikační protokol a strukturu JSON ke kódování dat.* Internetové stránky jako Google a Twitter shromažďují velké množství statických dat a dat z časových řad. Znalost API rozhraní těchto stránek umožňuje datovým analytikům a inženýrům přístup k velkému množství dat, která jsou na internetu neustále generována.

9.2.5.3 Senzory

Internet věcí (IoT) využívá k vytváření dat senzory. Tato data mohou pocházet z teplotních a vlhkostních senzorů používaných například v zemědělství. Senzory jsou nyní ve všem, od chytrých telefonů po auta a proudové motory, i v domácích spotřebičích. Ty spolu s mnoha dalšími typy senzorů (seznam věcí se senzory každým rokem roste) přispívají k exponenciálnímu růstu Big Data. Potřebujeme nové nástroje, nové technologie a nový způsob přístupu k tomu, jak ukládáme, zpracováváme a počítáme, aby se nezpracovaná data stala smysluplnou informací.

9.2.5.4 Databáze

Databáze obsahují data, která byla *extrahována*, *transformována* a *načtena* (ETL – extracted, transformed, loaded). ETL je proces „čištění“ nezpracovaných dat tak, aby je bylo možné umístit do databáze. Data jsou často uložena ve více databázích a pro analýzu musí být databáze sloučeny do jediné datové sady. Většina databází obsahuje data, která vlastní organizace a jsou soukromá. Existuje ale také mnoho veřejných databází, ve kterých může kdokoli vyhledávat. Ty mohou být k dispozici zdarma nebo za nízkou cenu.

9.2.6 PŘÍPRAVA DAT

9.2.6.1 Datové typy a formáty

Poté, co jsou data shromážděna z různých zdrojů, vyžadují přípravu na samotnou analýzu. Odborníci v oblasti Data Science odhadují, že příprava dat může zabrat 50 až 80 procent času potřebného k dokončení analýzy.

Vzhledem k tomu, že data mohou pocházet z různorodých zdrojů, nemusí být jejich kombinace nutně kompatibilní. Dalším problémem je, že data, která mohou být prezentována jako text, budou muset být převedena na číselný typ, pokud mají být použita pro statistickou analýzu. Datové typy jsou důležité, když se pro práci s daty používají počítačové jazyky, jako je **Python** nebo **R**. Některé různé typy dat a jejich popisy jsou zobrazeny na obrázku.

Data types

Data Type	Description
string	Data that is treated as text. It is composed of letters, numbers are not to be used in computation, and symbols, such as punctuation. Finally, string data typically includes white space, or the spaces used to separate and format text.
integer	Whole numbers, or numbers that don't include decimals. Depending on the computer language, integers may not include negative numbers. Integers are used to order things or in rankings.
floating point	Numbers with decimal places. These numbers are frequently employed in statistical analysis.
date and time	Important in recording when an observation in a data set was made. Date and time formats can vary widely between data sources.

Obr. Datové typy

Kromě různých datových typů může být stejný typ dat formátován odlišně v závislosti na jeho zdroji. Různé jazyky mohou například používat různé symboly k reprezentaci stejného slova. Britská angličtina může používat jiné hláskování než americká angličtina.

Formáty data a času představují výzvu. Přestože jsou čas a datum velmi specifické, jsou zastoupeny v široké škále formátů. Čas a datum jsou zásadní pro analýzu pozorování časových řad. Proto je nutné je převést do standardního formátu, aby měla analýza nějakou hodnotu. Data mohou být například v některých zemích formátována tak, že jako první bude následovat rok a den a měsíc, zatímco jiné země mohou prezentovat data s prvním měsícem, za nímž následuje den a rok. Podobně může být čas reprezentován ve 12hodinovém formátu s označením AM a PM nebo může být reprezentován ve 24hodinovém formátu. Různé formáty času a data jsou znázorněny na obrázku.

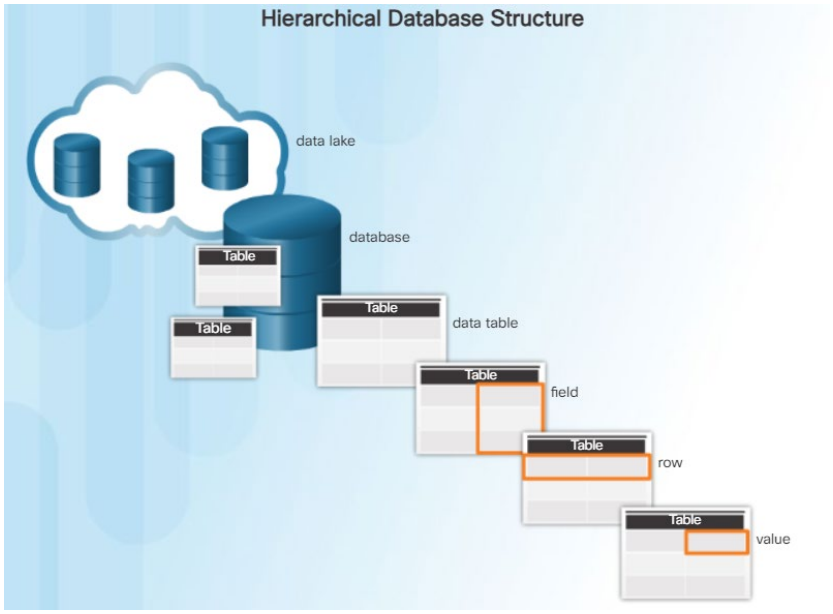
Data Formats

Standard	Representation	Example
ISO 8601	YYYY-MM-DDThh:mm:mm+|-ss:mm	2016-01-20T15:20:09
RFC 1123	DAY DD-MM-YYYY hh:mm:ss GMT|UT	Fri 20 Jan 2017 20:20:27 GMT
American	MM/DD/YYYY hh:mm:ss AM|PM	01/20/2016 3:20:09 PM

Obr. Datové formáty

9.2.6.2 Struktura dat

Databáze je kolekce datových tabulek, které spolu souvisí jedním nebo více způsoby. **Datové tabulky** se skládají z **polí**, **řádků** a **hodnot**, které jsou ekvivalentní **sloupcům**, **řádkům** a **buňkám** v tabulce. Každou datovou tabulku lze považovat za **soubor** a databázi za **kolekci souborů**. Obrázek znázorňuje vztah těchto struktur a související terminologii.



Obr. Hierarchická databázová struktura

Jiné datové struktury nebo objekty používá Python. Například Python používá jako primární datové struktury **řetězce**, **seznamy**, **slovníky**, **n-tice** a **množiny**. Každá datová struktura má svou vlastní skupinu funkcí nebo metod, které lze použít pro práci s objektem. Obrázek ukazuje běžné datové struktury Pythonu. Kromě toho oblíbená knihovna pro analýzu dat v Pythonu s názvem „pandas“ používá další datové struktury, jako jsou série a **datové rámce**.

Large Data Sets

Structure	Definition	Example
list	A series of numbers, characters, or strings separated by commas. Elements can be added, deleted or changed without the need to create a new list.	[1, 2, 3] ["apples", "oranges", "plums"]
tuple	Similar to lists, but their contents cannot be changed without creating a new tuple.	(1, 2, 3) ("apples", "oranges", "plums")

dictionary	A group of paired items called keys and values.	{
		"France":"French",
		"Italy":"Italian",
		"USA":"English"
		}

sets	An unordered set of unique items. If a set is created from a group of items that are not unique, only the unique items will form the set.	{1,2,2,3,5,7,7} = {1,2,3,5,7}

Obr. Velké datové soubory

9.2.6.3 Extrakce, transformace a načtení dat

Jak bylo již dříve zmíněno, velká část dat, která jsou umístěna do databáze, aby v nich bylo možné následně vyhledávat, pochází z různých zdrojů a je v široké škále formátů. ETL (extrahovat, transformovat a načíst) je proces pro sběr dat z této řady zdrojů, transformaci dat a následné načtení dat do databáze. Data jedné společnosti lze nalézt v dokumentech Word, tabulkách, prostém textu, PowerPointech, e-mailech a souborech PDF. Tato data mohou být uložena na různých serverech, které používají různé formáty.

1. Extrahovat – data jsou sbírána z několika zdrojů
2. Transformovat – agregace, třídění a čištění dat a spojování dat
3. Načtení – transformovaná data se poté načtou do databáze pro dotazování

Výše uvedené popisy tří kroků procesu ETL jsou zjednodušené. Ve skutečnosti je před načtením dat do databáze a následným dotazováním potřeba udělat poměrně hodně práce.

9.2.6.4 Extrahování dat

Krok extrahování shromažďuje požadovaná data ze zdrojů a zpřístupňuje je ke zpracování. Extrakce převádí data do jediného formátu, který je připraven k transformaci. Například zkombinováním dat ze serveru NOSQL a Oracle DB získáte data v různých formátech. Tato data musí být převedena do jediného formátu. Také je třeba zkontrolovat data, aby se zajistilo, že mají požadovaný typ informace (hodnotu). To se provádí pomocí validačních pravidel. Pokud data nesplňují pravidla, mohou být odmítnuta. Někdy jsou tato odmítnutá data opravena a následně ověřena znovu.

9.2.6.5 Transformace dat

Krok transformace používá pravidla k transformaci zdrojových dat na typ dat potřebný pro cílovou databázi. To zahrnuje převod jakýchkoli naměřených dat na stejnou dimenzi (např. imperiální na metrickou). Transformační krok také vyžaduje několik dalších úkolů. Některé z těchto úkolů jsou **spojování dat** z několika zdrojů, **agregace dat**, **třídění**, **určování nových hodnot**, které se počítají z agregovaných dat, a poté použití pravidel **ověřování**, **čištění dat**.

9.2.6.6 Načtení dat

Krok načtení nastane, když se transformovaná data načtou do cílové databáze. Může to být **prostý databázový soubor**, nebo **relační databáze**. Skutečný proces načítání se značně liší. Záleží na typu zdrojových dat, typu cílové databáze a typu dotazování, které má být provedeno. Některé organizace mohou přepsat existující data kumulativními daty. Načítání nových transformovaných dat lze provádět na hodinové, denní, týdenní nebo měsíční bázi. Může k tomu dojít pouze tehdy, když došlo k určitému množství změn v transformovaných datech.

9.2.6. PŘÍPRAVA NA CVIČENÍ: MĚŘENÍ RYCHLOSTI INTERNETU

9.2.6.1. Formátování data a času

Jak již bylo zmíněno, data IoT, která byla zkombinována z mnoha zdrojů, mohou být formátována způsoby, které jsou nekompatibilní. Existuje například mnoho způsobů, jak lze prezentovat údaje o datu a čase. Pro účely analýzy dat je však nejlepší, aby byly data a časy formátovány konzistentně. Jedním ze způsobů, jak se s tímto problémem vypořádat, je jednoduše použít data, která mají jediný formát času a data. To by však vedlo k tomu, že by analytik vyřadil relevantní, ale nekompatibilně zformátovaná data, což by vytvořilo zkreslení a vedlo k chybným závěrům.

Jeden z důvodů, proč je Python tak oblíbený u datových analytiků, je, že základní funkce jazyka byly rozšířeny o mnoho různých knihoven a modulů. Jeden modul Pythonu, který bude použit v nadcházejícím cvičení. Tento modul se nazývá datetime. Klikněte sem a přečtěte si podrobnou dokumentaci modulu datetime.

Modul **datetime** je součástí většiny distribucí Pythonu jako standardní knihovna, **musí být však importován**, aby mohl být použit ve vašem kódu. Vlastnosti modulu datetime jsou reprezentovány paradigmatem objektově orientovaného programování. **Modul se skládá z tříd datum, čas a datum a čas.** Každá třída má své **vlastní metody**, které lze volat pro práci s instancemi tříd nazývaných objekty.

Use the datetime module classes in your code:
import datetime as dt

Datetime Module Terminolgy

Term	Example	Use
module	datetime	import **datetime** as dt
class	time date datetime, etc.	dt.**time** dt.**date** dt.**datetime**
object	Variables t, d, and dateAndTime	**t** = dt.time(12,31,00) **d** = dt.date(1972,12,31) **dateAndTime** = dt.dateandtime.now()
method	strftime() weekday() isoformat	t.**strftime**(" %H:%M:%S") d.**weekday**() dateAndTime.**isoformat**()

Změna data a času z jednoho formátu do druhého se provádí pomocí metody **strftime** (řetězec z času), která je dostupná pro objekty **datetime**. Metoda strftime používá jako své parametry řadu formátovacích kódů nebo direktiv (viz tabulka).

Python datetime Formatting Directives

Directive	Meaning
%a	Locale's abbreviated weekday name
%A	Locale's full weekday name
%b	Locale's abbreviated month name
%B	Locale's full month name
%c	Locale's appropriate date and time representation
%d	Day of the month as a number [01,31]
%H	Hour (24-hour clock) as a number [00,23]
%I	Hour (12-hour clock) as a number [01,12]
%j	Day of the year as a number [001,366]
%m	Month as a number [01,12]
%M	Minute as a number [00,59]
%p	Locale's equivalent of either AM or PM
%S	Second as a number [00,61]
%w	Weekday as a number [00,61]
%W	Week number of the year (Monday as the first day of the week) as a number [00,53]. All days in a new year preceding the first Monday are considered to be in week 0.
%y	Year without century as a number [00,99]
%Y	Year with century as a number
%Z	Time zone name (no characters if no time exists)

Následující kód ukazuje kód Pythonu, který používá modul datetime k reprezentaci data a času ve formátu běžně používaném ve Spojených státech.

.

```
#load the datetime module as dt
import datetime as dt

#create a datetime object that contains the current time
currentDT = dt.datetime.now()

#view the value of currentDT
print(currentDT)
```

```
2017-02-22 20:44:14.037597
```

```
#create a new string object that contains the reformatted date and time
UDdt = currentDT.strftime('%b %d, %Y %I:%M %p')
#display the result
UDdt
```

```
'Feb 22, 2017 08:44 PM'
```

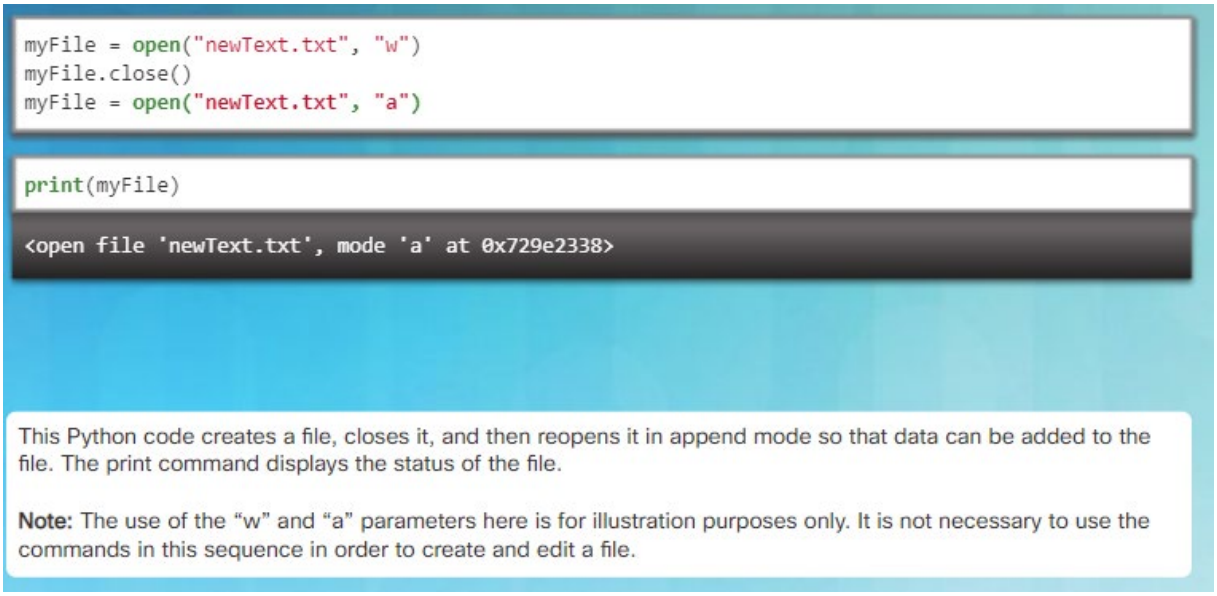
9.2.6.2 Čtení a zápis dat

Modul **csv** je také základní modul, který je součástí standardní knihovny Pythonu. Modul csv **umožňuje čtení a zápis do souborů .csv**.

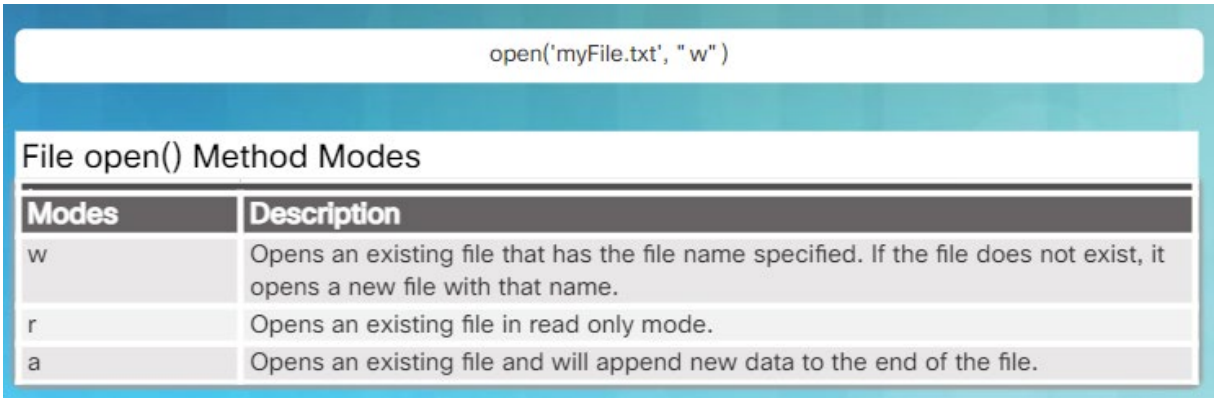
Python má také základní metody pro vytváření, otevírání a zavírání externích souborů. Metody Python open() a close().

Metoda **open()** se používá k vytvoření nového souboru nebo k otevření existujícího souboru, který bude obsahovat data k uložení.

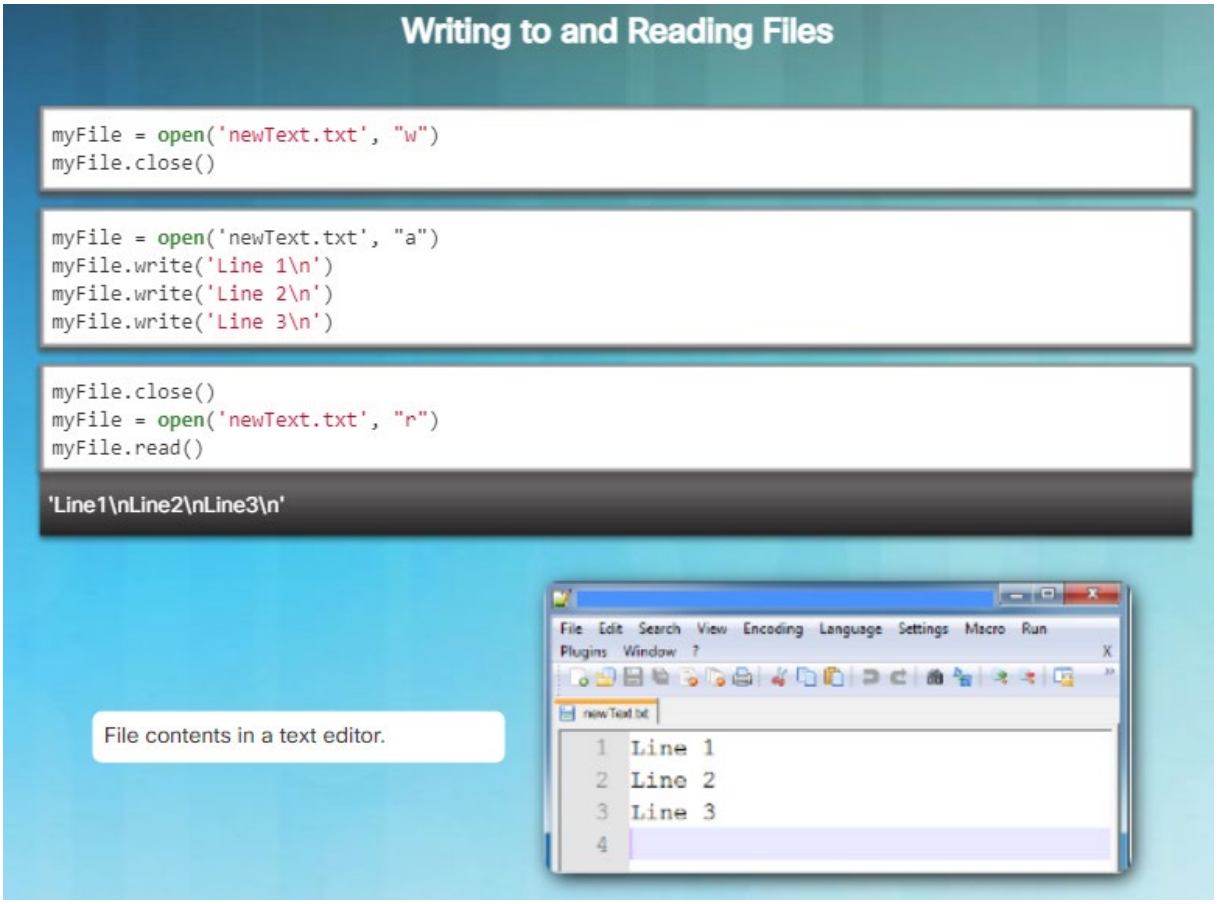
Funkce **close()** odstraní nezapsaná data z vyrovnávacích pamětí a ukončí funkci zápisu do souboru pro zadaný soubor. Je důležité výslovně zavřít všechny soubory, do kterých se nemá zapisovat. To šetří systémové prostředky a chrání soubor před poškozením.



Obrázek 2 vysvětluje některé důležité parametry, které lze uvést v metodě **open()**. Tyto parametry lze kombinovat nebo lze přidat symbol " " pro určení, že mají být použity režimy **čtení** a **zápisu** nebo čtení a **připojení**.



Data lze do souboru zapsat pomocí metody **write()**. Pokud byl soubor otevřen v režimu **„a“**, budou data přidána na konec souboru. Pro formátování může být nutné do souboru přidat **znaky escape** (někdy nazývané escape sekvence). Například znaky **\n** nebo **\r\n** přidají zalomení řádku na konec řádku dat, který byl zapsán. Souborová metoda **read()** čte obsah otevřeného souboru.



9.2.6.3 Interakce s externí aplikací

Python umožňuje interakci s externími aplikacemi a operačním systémem. V Jupyter Notebooku je symbol „!“ umožňující přímou interakci s operačním systémem.

```
! ls -al

total 80
drwxr-xr-x 4 root root 16384 Feb 22 19:47 .
drwxr-xr-x 5 root root 16384 Feb  6 22:34 ..
drwxr-xr-x 4 root root 16384 Feb 19 21:01 chapter 3
drwxr-xr-x 2 root root 16384 Feb 22 19:47 .ipynb_checkpoints
-rw-r--r-- 1 root root    72 Feb 22 19:47 Untitled.ipynb
```

```
!head logins.csv
```

```
Student,Login_Day,Login_Time
Rose,12/4/2016,20:29
Joe,9/4/2016,22:37
Jerry,2/20/2017,4:18
Lawrence,4/7/2016,13:17
Roy,6/24/2016,15:30
Robin,9/6/2016,20:55
Harry,8/12/2016,14:26
Mary,1/26/2017,1:46
Stephanie,12/31/2016,10:23
Tammy,5/25/2016,7:08
```

Další obrázek znázorňuje použití modulu **subprocess** pro komunikaci s externí aplikací a uložení výstupu příkazu vydaného této aplikaci v objektu Pythonu. Nejprve se vytvoří objekt typu řetězec, který bude obsahovat příkaz k odeslání programu. V tomto případě máme v úmyslu odeslat příkaz do nástroje ping, který je dostupný z prostředí Linuxu. Tento příkaz poté, po jeho rozdělení na jednotlivá slova, odešleme do programu pomocí metody **subprocess**. Nakonec výstup příkazu uložíme do proměnné a rozdělíme do řetězce. Poté můžeme obsah objektu zobrazit příkazem print a jednotlivé jeho prvky adresovat pomocí řetězcové indexace.

```
'''import the subprocess library which is necessary for communication with external
apps'''
import subprocess
#We now execute the ping process as of from the shell:

pingCmd = 'ping -c 127.0.0.1'
process = subprocess.Popen(pingCmd.split(), stdout=subprocess.PIPE)
'''creat an object to hold the output of the process and split the output elements
into a list'''

process_output = process.communicate()[0]
process_output = process_output.split()
#view the contents of the output object
print(process_output)
```

```
['PING'. '127.0.0.1', '(127.0.0.1)', '56(84)', 'bytes', 'of', 'data.', '64', 'bytes',
 'from', '127..0.0.1', 'icmp_seq=1', 'ttl=64', 'time=0.094', 'ms', '64', 'bytes',
 'from', '127.0.0.1', icmp_seq=2', 'ttl=64', 'time=0.052', 'ms', '---', '127.0.0.1',
 'ping', 'statistics', '---', '2', 'packets', 'transmitted', '2', 'received,', '0%',
 'packet', 'loss,', 'time', '999ms', 'rtt', 'min/avg/max/mdev', '=',
 '0.052/0.073/0.094/0.021', 'ms']
```

```
#view the first five elements of the list
 process_output[0:5]
```

```
['PING', '127.0.0.1', '(127.0.0.1)', '56(84)', 'bytes']
```

POUŽITÁ LITERATURA

[1] CISCO Big Data & Analytics: Advanced Data Analytics and Machine Learning. *Www.netacad.com* [online]. [cit. 2021-11-24]. Dostupné z: https://contenthub.netacad.com/legacy/IoTFBDA/2.01/en/index.html#4.0.1.1