Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Plzeň, 2018

Bc. Jan Cibulka

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 22.8.2018

_____

Poděkování

Za pomoc při zpracovávání diplomové práce chci poděkovat vedoucímu práce, panu Ing. Ivovi Punčochářovi, Ph.D., který mi dával rady pro zlepšení mnohých aspektů této diplomové práce. Dále bych rád poděkoval odbornému konzultantovi, panu RNDr. Jiřímu Mičkemu, který mi radil, jakým způsobem přistupovat ke zpracovávanému problému a podělil se o své znalosti databázových systémů. Další poděkování patří panu Ing. Janu Ménerovi za řízení projektu sběru dat ze servisů, bez kterého by tato práce být vypracována. Rád bych poděkoval společnosti OPENMATICS s.r.o. za důvěru a poskytnutí prostředků pro vypracování této práce.

Anotace

Tato diplomová práce se zabývá navržením a vytvořením systému pro ukládání a analýzu velkého množství dat na cloudové platformě Microsoft Azure. Práce obsahuje přiblížení konceptu Big Data, porovnání relačních a nerelačních úložišťových služeb a přiblížení nástrojů, které se využívají pro analýzu dat a vytváření prediktivních modelů. Práce je vytvořena ve spolupráci se společností OPENMATICS s.r.o., která se zabývá sběrem dat z automobilů. V práci je díky tomu zpracováván problém, ve kterém je třeba uložit jízdní a servisní data automobilů a na základě prediktivního modelu případně doporučit servis vozidla.

Klíčová slova

Big data, databáze, datová analýza, strojové učení, Microsoft Azure.

Annotation

This master thesis is focused on design and creation of solution for data storage and analysis on cloud platform Microsoft Azure. The thesis describes concept of Big Data, compares relational and non-relational databases and introduces tools, that are used for data analysis and predictive model creation. The thesis was created in collaboration with OPENMATICS s.r.o. which focuses on data collection from vehicles. Because of that, the principles described in the thesis are applied to an example from the automotive world. In the example, driving and service data are stored and a predictive model is created to determine if vehicle service will be recommended.

Keywords

Big Data, database, data analysis, machine learning, Microsoft Azure.

Obsah

Kapitola 1

Úvod

V této diplomové práci je popsáno využití moderní cloudové platformy pro ukládání velkého množství dat, jejich analýza a vytváření prediktivních modelů pomocí strojového učení. Práce je realizována ve spolupráci se společností OPENMATICS s.r.o., která používá cloudové prostředí Microsoft Azure s širokým spektrem nástrojů a služeb. Předmětem analýzy jsou data, která jsou získávána z automobilů a automobilových servisů v podobě zpráv, odesílaných do cloudového prostředí, kde jsou data uložena, zpracována a prezentována pomocí vizualizací.

Společnost OPENMATICS se zaměřuje na vývoj zařízení, která jsou nainstalována do nákladních a osobních automobilů. Zde sbírají jízdní a diagnostická data, která jsou posílána do cloudového prostředí, kde jsou ukládána a zpracovávána. V této práci jsou jízdní data kombinována se servisními daty vozidel, která jsou sbírána nově vyvinutým zařízením ZF Smart Device. V práci je navržen a realizován konkrétní příklad ukládání a analýzy dat s reálným vzorkem automobilů. V tomto příkladu jsou získaná data využita ke trénování klasifikátoru pomocí strojového učení. Tento klasifikátor ze stavu vozidla a stylu jízdy odhaduje stav brzd a zda je třeba provést servisní kontrolu.

Práce je rozčleněna do dvou hlavních částí. První část se věnuje extrakci, transformaci a nahrávání dat do zvoleného úložiště. Tento proces se označuje jako ETL (Extract, Transform, Load). V této části je popsán rozdíl mezi relačním a nerelačním úložištěm dat a uveden přehled často používaných cloudových služeb představujících tato úložiště. Jsou zde popsány parametry jednotlivých typů úložišť. Na základě tohoto rozboru je vybráno a vytvořeno úložiště, které je výhodné pro uchovávání a analýzu velkého objemu dat.

Ve druhé části práce je uveden přehled nástrojů využívaných pro analýzu uložených dat a vytváření vizualizací získaných výsledků, které mohou být využity při operativním

i strategickém rozhodování. Data získaná z analýz jsou použita ve strojovém učení pro trénování klasifikátoru. Je využívána služba Azure Machine Learning, která poskytuje vhodné nástroje pro přípravu dat, trénování a používání natrénovaného klasifikátoru. Na závěr práce jsou prezentovány vizualizace, na kterých jsou vyobrazeny různé metriky a analýzy uložených dat. Je vytvořeno několik reportů, ve kterých jsou jízdní data ukázána na mapě a v časovém grafu rychlostí pro jednotlivá vozidla, panel pro vizualizaci identifikovaných jízdních událostí a vizualizace činnosti klasifikátoru.

Kapitola 2

Představení produktů firmy, prostředí Azure a konceptu zpracování velkého objemu dat

2.1 Představení firmy a jejích produktů

2.1.1 OPENMATICS

Společnost OPENMATICS s.r.o byla založena v Plzni v roce 2010. Je součástí průmyslového koncernu ZF Friedrichshafen, který se zaměřuje na špičkové technologické vybavení zejména v automobilovém a přepravním průmyslu.

Firma cílí na vývoj inteligentních zařízení, které usnadňují sledování a správu v oblasti přepravy a logistiky. Cílem společnosti je vytvoření otevřené telematické platformy, která bude využívána různými přepravními a logistickými společnostmi. OPENMATICS vyvinulo několik variant zařízení pro sběr dat ze senzorů vozidla, které jsou následně ve formě komprimovaných zpráv odesílány do zvoleného úložiště. Dvěma základními variantami sběrných zařízení jsou Bach Box a Vivaldi Box. Bach Box je ukázán na obrázku 2.1 a je určen pro nákladní vozidla a autobusy. Pro osobní automobily je určen Vivaldi Box.

Společnost OPENMATICS vyvíjí inteligentní aplikace na míru, které z uložených záznamů vozidla získávají klíčové informace pro zákazníka, například sledování spotřeby paliva ve vozidlech nebo hlídání dodržování povinných přestávek řidičů nákladních vozidel. Zákazník má své informace k dispozici například formou portálové aplikace nebo na panelu v autobusu.

Obrázek 2.1: Zařízení Bach Box určené pro sběr dat v nákladních vozidlech.

Kromě aplikací, které sledují aktivitu přepravní flotily a optimalizují její chod, vytváří společnost produkty pro sledování přepravních zásilek v areálu skladu, vysílání obsahu pro cestující do autobusů a sběr diagnostických dat, které včas odhalí blížící se selhání součástky ve vozidle.

2.1.2 Vivaldi Box

Sběrné zařízení na obrázku 2.2 nese název Vivaldi. Připojueje se v prostoru řidiče do OBD sběrnice, která je součástí osobních automobilů od roku výroby 2000. Zařízení Vivaldi je určeno pro odesílání standardních telematických a diagnostických údajů o vozidle. Zařízení obsahuje SIM kartu, přes kterou se připojuje k datové síti, pokud není k dispozici síť Wi-Fi. Mezi další vybavení patří GPS senzor, Bluetooth modul, teploměr a jako vnitřní úložiště používá SD kartu. Signály, které je možné číst, jsou různé dle typu vozidla, zatím je však definováno 74 různých jízdních signálů v autě, které mohou být snímány. Dále je snímána sada signálů, které hlásí stav součástek, varování a popřípadě závady. Data jsou poté ukládána do cloudových úložišť a je s nimi nakládáno podle přání zákazníka.

2.1.3 ZF Smart Device

ZF Smart Device je zařízení, které svým vzhledem připomíná Vivaldi Box. Není však určeno pro sběr jízdních dat, ale ke sběru dat při vykonávání servisní úpravy. Technik v servisu má spárovanou mobilní aplikaci se ZF Smart Device zařízením které je při servisním úkonu připojeno k OBD sběrnici vozidla. V mobilní aplikaci vybere jednu z pěti podporovaných akcí:

- Nastavení hlídání intervalu příští návštěvy servisu (Reset Service Indicator),
- výměna brzdových destiček,

Obrázek 2.2: Zařízení s názvem Vivaldi Box, určené pro sběr dat v osobních vozidlech a mobilní aplikace, která obsahuje vizualizaci těchto dat.

- odvzdušnění brzdového systému,
- přečtení chybových kódu na OBD sběrnici vozidla (EOBD Diagnostics),
- a kalibrace úhlu volantu.

Mobilní aplikace provede technika doporučenými kroky při servisním úkonu. Technik do ní zadá informace o stavu servisní úpravy. Ty se synchronizují s vozidlem přes OBD a zároveň se odesílají na OPENMATICS cloud, kde se ukládají a dále zpracovávají. Návrh uložení a analýzy dat z těchto servisních úkonů je hlavním cílem této diplomové práce.

2.2 Zpracování velkého objemu dat

V této části je přiblížen koncept systému určeného pro ukládání, zpracovávání a analýzu velkého objemu dat (v angličtině 'Big Data'). V následujících kapitolách jsou podrobněji analyzovány varianty datového úložiště a procesních nástrojů.

2.2.1 Co jsou Big Data

Jak je uvedeno v článku [9], Big Data je obecný pojem pro strategie a technologie využité pro sběr, organizaci, zpracování a získání nových poznatků z velkého objemu dat.

S rozšiřováním Internetu, cloudových služeb a spektra zařízení, která mohou být použita ke sběru dat, narůstá i možnost využití Big Data. Dříve byla data ukládána

izolovaně a sloužila obvykle jednomu specifickému účelu. Navíc bylo ukládáno pouze nezbytné množství dat kvůli vysokým cenám úložišť. Dnes jsou velké objemy dat přístupné z veřejných zdrojů, ze sociálních sítí, Internetu věcí (IoT) a jiných online služeb, které poskytují svá data k dalšímu zpracování. Cena úložišť stále klesá a objevují se aplikace využívající stále širší spektrum dat.

2.2.2 Charakteristika Big Data

V mnoha zdrojích, například [18], jsou Big Data charakterizována pomocí tří faktorů, Volume (Objem), Velocity (Rychlost) a Variety (Rozmanitost), které se označují jako '3V'.

Volume Koncept Big Data je založen na použití agregací dat z mnoha různých zdrojů, které produkují velké množství dat strukturovaného i nestrukturovaného typu. Čím více dat, tím přesnější je datový model a tím více možností existuje při datové analýze. Devadesát procent dat, která jsou dnes k dispozici, bylo vytvořeno v posledních dvou letech.

Velocity Další charakteristikou Big Data je rychlost, kterou jsou data streamována a zpracovávána. Mnoho dat, například ze sociálních sítí, je vytvářeno velmi rychle a data jsou relevantní pouze omezenou dobu. Každou minutu je nahráno 72 hodin záznamů na YouTube, 216 000 nových fotek na Instagramu a posláno 204 000 000 emailů.

Variety Big Data obsahují data z mnoha vnějších zdrojů, které mají různé způsoby reprezentace dat. Je třeba umět zpracovávat tradiční databáze, audio, video, textové dokumenty a jakýkoliv jiný přípustný zdroj dat.

Veracity Důvěryhodnost a kvalita dat jsou také důležité charakteristiky Big Data řešení. Odhadovaná škoda na ekonomice Spojených Států za každý rok způsobená nízkou kvalitou dat je 3.1 miliardy dolarů.

2.2.3 Výhody plynoucí z Big Data

Rozšiřování Big Data vedlo ke zdokonalování platforem zaměřených na analýzu velkého objemu dat a prediktivní diagnostiku systémů pomocí algoritmů strojového učení.

Největší výhody vyplývají právě z '3V', které byly zmíněny. Společnost, která dokáže v reálném čase zpracovávat velké množství dat z různých zdrojů, tak může vytvořit řešení, které dokáže předvídat události s dostatečnou přesností a rychlostí a generovat užitečná a včasná rozhodnutí.

2.3 Popis prostředí MS Azure

2.3.1 MS Azure

Microsoft Azure je cloudová platforma, která poskytuje celou řadu služeb a nástrojů pro ukládání a analyzování rozsáhlých dat. První verze této platformy byla představena v roce 2008 společností Microsoft. Přehledné portálové rozhraní umožňuje vytvářet aplikační řešení pro firmy i jednotlivce. Na Azure jsou k dispozici veškeré nástroje pro vývoj, testování, běh i monitorování těchto řešení. Microsoft provozuje na světě zhruba dvě desítky výpočetních center, kde jsou fyzicky uživatelské aplikace provozovány.

Základní charakteristiky MS Azure

Snadná škálovatelnost

Aplikace mají možnost "nafouknout" použité komponenty tak, aby byl zajištěn plynulý chod při nárazovém zvýšení počtu zpráv či obejmu dat. Například když je vytvářena komponenta, která z prvního úložiště čte zprávy automobilů, transformuje je a ukládá do dalšího datového úložiště, tak přes noc není potřeba ani zdaleka tak vysoký výpočetní výkon jako přes den. Je možné nakonfigurovat spouštění více instancí aplikace, pokud se zvýší odezva služby. Škálovatelnost funguje i směrem dolů, jedná se o tzv. downsizing. Není totiž třeba si pořizovat dvakrát větší úložiště, než bude ve skutečnosti potřeba, pouze pro pocit bezpečí.

Úložiště

Jedna z hlavních služeb poskytovaných na MS Azure jsou datová úložiště. Na MS Azure je snadné navyšovat kapacitu úložišť nebo nakonfigurovat třídění dat na více míst. Součástí datových úložišť na Azure je i možnost geografické redundance, která zaručuje replikování dat mezi více datacentry. V praxi to znamená, že data jsou automaticky zálohovaná a chráněna pro případ, kdy je vyřazeno z provozu jedno datacentrum. Ceny za úložný prostor jsou také levnější než u běžných webhostingů. Velkou výhodou je i ukládání dat do regionu, kde dochází k využívání dat, což dále snižuje cenu služby.

Zabezpečení

Autorizace přístupů ke komponentám a zabezpečení je další výhoda v MS Azure. Přihlašování probíhá přes Microsoft účty. Každá komponenta má definované seznamy uživatelů s různými právy. Při používání internetového prohlížeče tedy může aplikace vyžadovat přihlášení, které je jednoduchým způsobem jak pracovat s přístupností obsahu pro různé skupiny uživatelů.

Portálové rozhraní

Azure nabízí správu komponent přes uživatelsky přívětivé prostředí Azure portálu. Je možné vytvářet, spravovat a monitorovat aplikace kdekoliv na světě pouze z jednoho webového portálu, který je interaktivně propojen s dokumentací a dalšími informacemi, které jsou při vývoji potřebné.

Integrace do Visual Studia

Azure je integrováno do Visual Studia, které je oblíbeným vývojářským studiem a nabízí podporu mnoha jazyků a modulů pro vývoj a vzdálený monitoring běžících komponent v debug módu. Často využívanými vývojářskými moduly je například SQL Server Data Tools (SSDT), který se využívá při návrhu databází a analytických funkcích, nebo development SAJ (Stream Analytics Job), které umožňují zpracovávání dat v reálném čase.

2.3.2 Používané komponenty v MS Azure

Základním stavebním kamenem cloudového systému vytvořeným v Azure jsou komponenty, které představují typ služby, která je od Microsoftu pronajímána pro dané uživatelské prostředí a slouží konkrétnímu účelu.

Existuje mnoho typů komponent, které je možné využít v prostředí MS Azure. V následujícím seznamu jsou uvedeny nejčastější kategorie, které se zde využívají a ke každé je uvedeno pár příkladů konkrétní komponenty.

Compute

Azure nabízí pronájem výpočetního výkonu. Může se jednat o malé jednotky, tzv. Azure Functions, které se hodí pro psaní malých skriptů v jazyce C# nebo Python, které se vykonávají třeba po příchodu zprávy a ukládá jí do databáze nebo se může jednat o pronajmutí sady počítačů s velkým výkonem pro numerické výpočty.

Networking

Jsou zde nástroje pro tvorbu, monitorování a směrování ve virtuálních sítích.

Úložiště

Může se jednat o tabulky, souborová úložiště, disková úložiště, fronty se zprávami, SQL databáze nebo Azure Data Lake Store, který umožňuje výhodné ukládání jakéhokoliv formátu dat.

Web Na Azure je kompletní podpora hostování webových aplikací, mobilních aplikací, pronajímání veřejných internetových adres a možnost tvorby vlastního webového rozhraní (API).

Analýza dat

Je zde možné využívat laboratoř strojového učení, kognitivní nástroje jako strojové vidění nebo rozpoznávání řeči a v neposlední řadě rychlá tvorba chytrých reportů nad uloženými daty.

IoT Internet věcí (Internet of Things) je stále rostoucí odvětví ve využití potenciálu moderních technologií a Azure nabízí všechny nástroje potřebné pro jeho vývoj a běh. Poskytuje bránu, přes kterou se zařízení připojují, Event Hub, kde se hromadí zprávy z IoT zařízení čekající zpracování a nástroje pro třídění a trvalé ukládání zpráv.

Stream Processing

Tyto komponenty se zaměřují na zpracování dat v reálném čase. Může se jednat o nahrávání souborů, zpracovávání zpráv nebo přesměrovávání toku dat do jiných zpráv, souborů nebo databází. Zpracovávání dat zde zahrnuje dotazování, filtrování a shlukování dat v obrovském objemu a s co nejnižší dobou odezvy.

2.3.3 Big Data na MS Azure

Každý systém pro Big Data analýzu se skládá z několika komponent. Je potřeba používat technologie pro zpracování datových zdrojů, ukládání těchto dat do datového úložiště, využívat takto uložená data pro tvorbu datových modelů a prezentování získaných informací ve vizualizacích a reportech. MS Azure nabízí všechny komponenty, které jsou potřeba k vytvoření a provozu Big Data řešení.

Koncept Big Data řešení se skládá ze několika úrovní, které jsou graficky znázorněné na obrázku 2.3.

Zdroje Dat

První úrovní jsou zdroje dat. Je třeba nakonfigurovat brány, kterými data vstupují do systému, jsou tříděna a ukládána do příslušných úložišť. Tato úložiště jsou poté používána jako datové zdroje pro následnou analýzu. Data mohou být ukládána ve strukturovaném nebo nestrukturovaném formátu.

Zdroj může být například internetová aplikace, logy uložené na disku nebo údaje ze senzorů ve vozidle.

Jako vstupní brána pro data ze senzorů může být využita komponenta IoT Hub. Na vstupu IoT Hub jsou komprimované zprávy z externích zařízení. Ty se v IoT Hub roztřídí a přepošlou do dílčích cloudových služeb, které zprávy připraví pro uložení v datovém úložišti.

Datová úložiště jsou široký pojem a této problematice se věnuje následující kapitola. Hlavní funkcí je pružně ukládat obrovské množství dat. Úložiště musí splňovat požadavky na velikost, flexibilitu a rychlost využívání. Také musí být kompatibilní s nástroji, které integrují data do analytických nástrojů. Základními komponentami, které se využívají pro ukládání nestrukturovaných dat, jsou Azure Data Lake, Azure Storage Account a Azure Cosmos DB. Pro strukturovaná data se využívají relační databáze, nejčastěji Azure SQL Database a Azure Data Warehouse.

Integrace Dat

Nad datovým úložištěm je třeba vytvořit službu, která se stará o optimální distribuci dat v datovém úložišti, paralelní zpracovávání dat a následné analýzy. Využívají se frameworky od firmy Apache, například Hadoop, Spark a Hive, které poskytují nástroje pro efektivní mapování dat v úložišti a jejich zpracovávání.

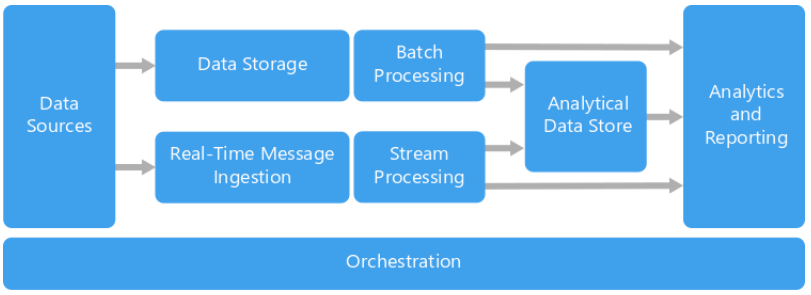
Datová analýza

Když jsou data připravená pro zpracování, je možné použít mnoho různých služeb pro jejich analýzu. Základní službou pro analýzu dat z relační databáze je Azure Analysis Services. Ta poskytuje nástroje pro zpracovávání dat uložených v SQL databázích, agregování a filtrování dat a počítání metrik, které slouží ke shrnutí uložených dat. Pro nestrukturovaná data slouží analytické nástroje v Azure Data Lake Analytics komponentě. Další často využívanou komponentou je HDInsight, která využívá integraci dat pomocí výpočetní platformy Apache Spark a umožňuje zde běh různých analytických nástrojů. Příkladem je platforma pro statistické výpočty v jazyce R, prostředí pro návrh a trénování strojového učení v Pythonu (Jupyter Notebook) nebo využití Kognitivního Toolkitu od Microsoftu pro analýzu obrazu nebo řeči.

Vizualizace a Reporting

Poslední úrovní Big Data řešení je efektivní prezentace získaných výsledků. Zde je třeba vybrat službu, která nejvíce odpovídá požadavkům, které jsou na řešení kladeny. Požadované výstupy mohou být tabulky, grafy, interaktivní vizualizace nebo natrénované modely klasifikátorů. Je možné ukázat získaná data za použití SQL Server Reporting Services nebo extrahovat data do Microsoft Excel, který se překvapivě stále často využívá pro prezentaci výsledků.

Pro tvorbu inteligentních dashboardů je vhodné využít platformu Microsoft Power BI. Natrénovaný model strojového učení může být k dispozici pomocí webové služby. Pro ukládání, sdílení a uspořádání informací je možné využít Microsoft SharePoint, který vytváří weby pro organizace, optimalizované i pro přístup z mobilních zařízení.



Obrázek 2.3: Struktura Big Data řešení, které je zaměřeno a real-time streamování dat do vizualizací a dávkové zpracování uložených dat v datovém modelování. Zdroj: [24]

Po popsání základních úrovní, které je třeba zajistit při návrhu Big Data řešení, budou popsány další nástroje, které je možné využít.

2.3.4 Machine Learning na MS Azure

Pro navrhování, trénování a správu běhu modelů strojového učení je možné použít systém Azure Machine Learning (AML), který je tvořen komponentami Azure Machine Learning Experience která se stará o vytváření modelů strojového učení, a Azure Machine Learning Model Management, kde jsou tyto modely registrovány a využívány pomocí webových služeb. Podrobnější popis AML je uveden v části 5.2

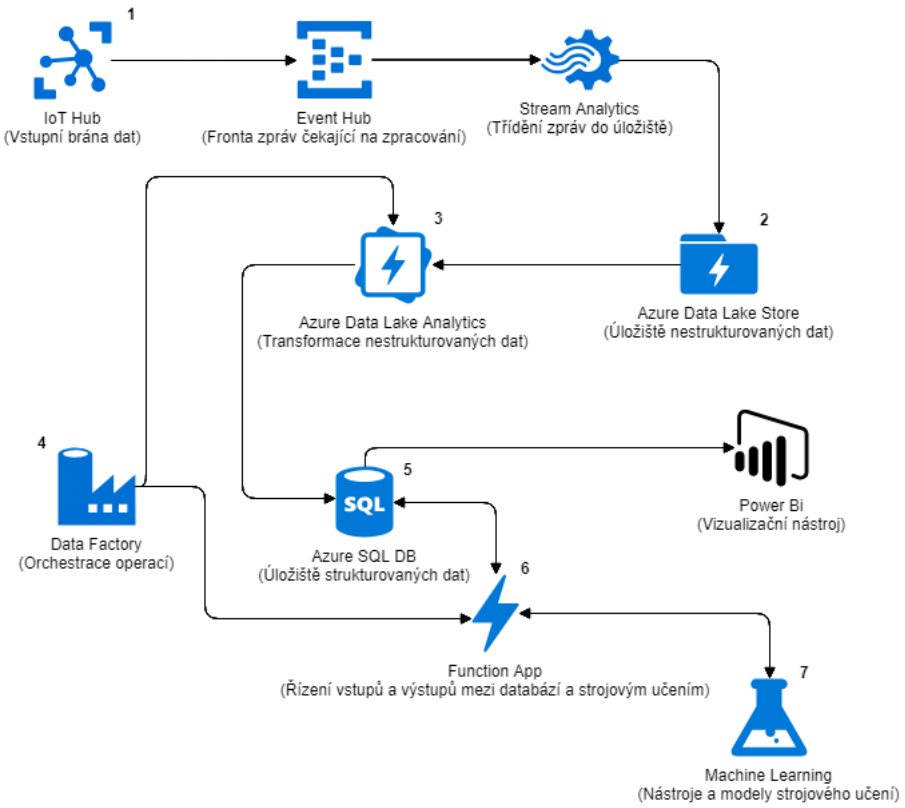
Další možností pro vytvoření aplikace strojového učení je použití Machine Learning Studia, které umožňuje velmi rychlé a intuitivní vytváření modelů. Tyto varianty budou popsány v následujících kapitolách.

2.3.5 Řízení běhu Big Data řešení

Celá struktura musí být něčím řízena, aby do sebe zapadaly jednotlivé akce. Řídí se opakované načítání dat, zapouzdření akcí do úloh, jejich synchronizace a distribuce výsledků analýzy do zvolených výstupů systému. K těmto akcím se používá Azure Data Factory, která obsahuje nástroje a rozhraní pro správu datových toků a časování úloh.

2.3.6 Ukázková architektura Big Data řešení

Schéma komponent na obrázku 2.4 je výhodné pro Big Data řešení.



Obrázek 2.4: Architektura vhodná pro pokročilou analýzu Big Data. Jsou zde uvedeny konkrétní komponenty, které je možné využít.

Jednotlivé komponenty na obrázku 2.4 mají následující funkce:

1. IoT Hub slouží jako vstupní brána dat ze zařízení do cloudového systému.
2. Data jsou tříděna a ukládána do úložiště nestrukturovaných dat Azure Data Lake Store. Více o této komponentě v části 3.4.2.2.
3. Data uložená v Azure Data Lake Store jsou analyzována pomocí komponenty Azure Data Lake Analytics. Více o této komponentě v části 5.4.2.
4. Azure Data Factory slouží k časování a řízení toku dat z nestrukturovaného úložiště do relačního modelu v SQL databázi. Po aktualizaci dat v SQL databázi také spouští Function App z bodu 5.
5. Azure SQL Database obsahuje relační model, ve kterém jsou ukládána strukturovaná data. Jsou zde tabulky s jízdními, servisními a referenčními daty. Pomocí

analytických nástrojů jsou zde vytvářeny statistiky pro jednotlivá vozidla a spravovány vstupní a výstupní data klasifikačního modelu.

6. Fucntion App obsahuje skript, který z SQL databáze získá vstupní data pro klasifikaci a předá je službám, které se starají o běh klasifikačního modelu. Výstupní data z těchto služeb pak zapíše do příslušné tabulky v relační databázi.

7. Azure Machine Learning obsahuje služby potřebné pro vytváření, správu a běh modelů strojového učení. Více o Azure Machine Learning v čísti 5.2.

Kapitola 3

Analýza ukládání dat na cloudu

V předchozí kapitole je představena architektura systému pro zpracování Big Data jako celek. V této kapitole jsou popsány varianty datových úložišť včetně jejich charakteristiky, scénářů využití a jejich výhody i nevýhody. Na konci kapitoly jsou vybrány služby MS Azure, které budou používány pro ukládání dat.

3.1 Strukturovaná, semi-strukturovaná a nestrukturovaná data

Prostředí Azure nabízí mnoho variant ukládání dat [15]. Hlavní rozdíl mezi nimi je použité datové schéma, ve kterém jsou data ukládána.

Strukturovaná data
: Data, která se řídí datovým modelem, ve kterém jsou definovány vztahy mezi modelovanými entitami. Strukturovaná data jsou ukládána po řádcích do relační databáze.

Nestrukturovaná data
: Data, která nejsou uložena do datového modelu. Nestrukturovaná data jsou ukládána po blocích do úložiště bez relačního schématu, obsah dat není zpravidla tříděn podle obsahu, ale podle časové značky záznamu a datového zdroje. Často se tato data označují jako 'Blob', odvozené ze zkratky termínu 'Binary Large OBject', který představuje blok dat.

Semi-strukturovaná data
: Kompromis mezi oběma přístupy. Částečně strukturovaná data obsahují značky nebo oddělovače různého obsahu v rámci dat. Někdy se označují jako "NoSQL"data, protože se ukládají do úložišť, která umožňují oddělovat části

obsahu, ale není vytvořen model, který by popisoval vlastnosti mezi uloženými entitami.

Podstatným rozdílem mezi relačními a nerelačními úložišti je fáze, kdy je aplikováno schéma na uložená data. U relační databáze je nutné vytvořit schéma dat již při zápisu. Relační databáze nedovoluje zápis, pokud nejsou všechna povinná pole vyplněna správným datovým typem. Snadno se tímto způsobem udržuje konzistence dat a také při čtení z relační databáze odpadá starost o to, zda jsou data ve správném formátu se správnými datovými typy atd., protože struktura je již zaručena schématem databáze.

Naopak v případě nerelačních databází se schéma aplikuje na data až při výstupu. Do nerelačního úložiště je možné nahrát v podstatě jakýkoliv obsah, který nemá jednotnou strukturu. V jednom ukládacím kontejneru se tak může mísit mnoho schémat vstupů.Tato výhoda je vykoupen tím, že čtecí mechanismus musí zajišťovat nástroje, které zajistí integritu, selekci a validaci dotazovaných dat.

3.2 Parametry datových úložišť

Níže je uveden seznam parametrů, které je vhodné zvážit při výběru typu datového úložiště.

Struktura dat

Jak složitá jsou data, se kterými se pracuje? Nestrukturovaná data jako např. média jsou ukládány do blobového úložiště, naopak strukturovaná data je možno třídit do tabulek v SQL databázi a vytvářet mezi nimi reference a relace.

Dotazování Dat

Jak složité dotazy úložiště používá? Zde mají relační databáze velkou výhodu, protože používají pro dotazovaní dat standardizovaný jazyk SQL. Naproti tomu úložiště typu klíč-hodnota je vhodné pro získávání jednotlivých záznamů, nalezení spojitostí mezi daty je však v tomto případě složitější a vyžaduje více kroků.

Škálovatelnost

Jaké jsou objemové limity pro využívání úložišťových služeb? Nestrukturovaná úložiště se zpravidla zvětšují automaticky, SQL server má limity a změna předplatného nebo roztřídění dat mezi více instancemi (database sharding) může vyžadovat pozornost a práci.

Integrita dat

Jak hodně je důležité, aby byla zaručena integrita dat? Nerelační úložiště

zpravidla nekontrolují dodržení datového schématu obsah, zato ale mají lepší poměr cena/výkon. Relační databáze mají zaručenou konsistenci obsahu, zpravidla výměnou za horší poměr ceny/výkonu.

Cena Kolik stojí vybrané úložiště? Pro každý typ existuje více modelů placení. Většinou je zpoplatněno více věcí, například celkový objem uložených dat nebo rychlost zápisu/čtení.

3.3 Úložiště strukturovaných dat

V úložišti strukturovaných dat [16] jsou data uložena podle relačního modelu, ve kterém je každý záznam reprezentovaný uspořádanou n-ticí hodnot a záznamy jsou strukturovány do tabulky po řádcích. Každý sloupec mám přiřazen význam a datový typ části záznamu. Jeden ze sloupců obsahuje povinný unikátní index, který umožňuje jednoznačnou identifikaci řádku tabulky. Tento model se často označuje jako "ER Model" (Entity-Relation Model), který je tvořen z tabulek, které jsou též nazývány entitami a představují prvky reálného světa, a relací, které představují vztahy mezi těmito entitami.

Pro přístup do relačních databáze se využívá jazyk SQL (Structured Query Language). Tento jazyk pouze definuje přípustné operace nad databází, ale již nespecifikuje, jakým způsobem se akce vykoná. Tato činnost je vyhrazena vnitřnímu engine databáze, který se postará o bezpečnost a efektivitu operace. Pokud se například někdo snaží číst i zapisovat najednou ten samý záznam. Provedení implementace tohoto engine je pak definována normou ISO/IEC 9075 [3].

Pro zajištění rychlosti a optimalizace zpracovávání dotazů se používají indexy. Primární (unikátní index) a sekundární index, který určuje rozložení dat v tabulce. Správná optimalizace těchto indexů může velmi zrychlit vykonávání dotazů nad databází. Využití relační databáze naráží na omezení, když je relační model příliš složitý. Při desítkách tabulek a ještě více relacích je v první řadě obtížné se v modelu vyznat. Protože může vložení jednoho vstupu ovlivňovat mnoho jiných tabulek, zajištění integrity a konzistence uložených dat musí být zajištěno vnitřním engine databáze.

Velkou výhodou relační databáze je zaručení integrity dat, protože z její podstaty není možné vložit data, která by nevyhovovala datovému modelu. Při provádění databázové transakce z jednoho konzistentního stavu do druhého musí databázová transakce splňovat tzv. vlastnosti ACID [1] (Atomicity, Consistency, Isolation a Duralibity).

Atomicita

Transakce je nedělitelná, provede se jako celek nebo se neprovede vůbec.

Konzistence

Při a po provedení transakce není porušeno žádné integritní omezení

Izolovanost

Operace uvnitř transakce jsou skryty před vnějšími operacemi. Pokud by bylo třeba vrátit transakci (rollback), nebude zasažena jiná transakce.

Trvalost

Výsledky úspěšných transakcích, jsou bezpečně uloženy v databázi.

Mezi nejčastěji využívaná datová úložiště strukturovaných dat pro Big Data řešení patří komponenty Azure SQL Database a Azure Data Warehouse.

3.3.1 Azure SQL Database

Azure SQL Database je služba, která slouží jako relační datové úložiště, které může obsahovat více databází. Databáze obsahuje ER Model naplněný daty, kontrolu přístupu k datům a jiné bezpečnostní prvky a programovatelné databázové funkce. Cena za pronájem služby se odvozuje od jednotek DTU konkrétní Azure SQL Database.

DTU [14] je obecná jednotka pro určení výkonu databáze, ve které je zastoupen výkon CPU, využití operační paměti na databázovém serveru a objem toku dat při čtení a zápisu. Azure nabízí mnoho úrovní DTU, od základní databáze s 5 DTU za 5$ až po 4000 DTU za 16,000$ měsíčně. Microsoft na svých webových stránkách [4] nabízí kalkulačku DTU, která pomáhá vybrat optimální velikost zakoupeného množství DTU. Pro ilustraci: 10 000 operací za sekundu využívá 8 jader procesoru a to odpovídá 1000 DTU. SQL Databáze mají omezení, u Premium verze je to 4TB kapacity a 4000 DTU.

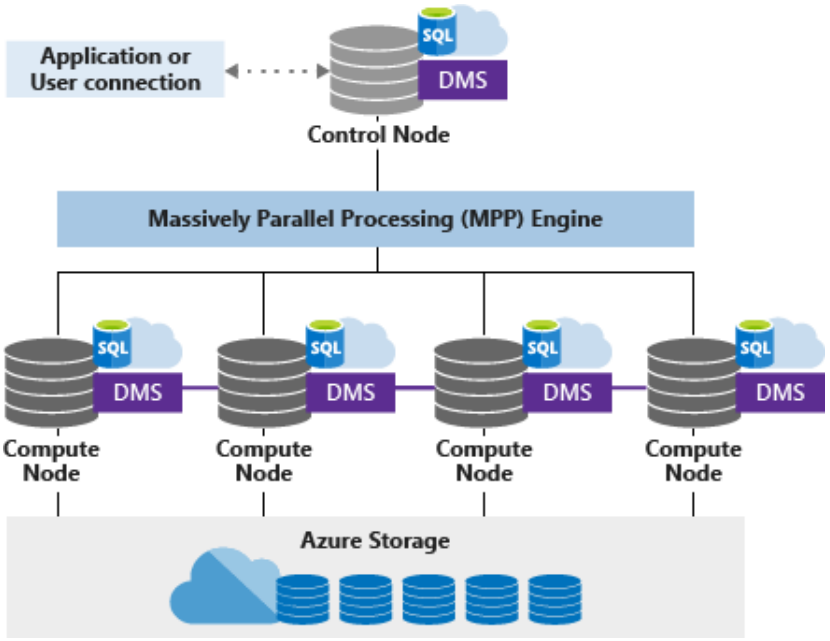
Při velkém objemu dat lze SQL databáze rozštěpit (sharding) a pomocí knihovny ElasticDatabase spravovat horizontální rozdělení dat mezi jednotlivými instancemi databází. Pro dotazování shardovaných databází se používá jazyk T-SQL (Transact SQL), což je jazyk SQL obohacen o funkce, procedury, indexy a požadované transakce mezi shardovanými databázemi. Při dotazování je možné využít strukturovanost dat a relací mezi tabulkami a tímto způsobem shlukovat informace (GROUP BY, JOIN) a mít zaručenou konsistenci a integritu dat. Pro design a spravování modelů v Azure SQL Database je často používána aplikace Microsoft SQL Server Management Studio, kde je možné snadno a rychle manipulovat s daty a navrhovat Entity Relation schémata.

3.3.2 Azure SQL Data Warehouse

Azure SQL Data Warehouse [8] je cloudový datový sklad strukturovaných dat, který používá technologii MPP [17] (Massively Parallel Processing) k realizaci složitých

dotazů nad velkým objemem dat. Schéma MPP je ukázáno na obrázku 3.1. Data jsou ukládána v relačních tabulkách v sloupcovém formátu.

Sloupcový formát je opačný od úložišť které jsou orientována na ukládání po řádcích, například úložiště klíč-hodnota (viz kapitola 3.4.1) a optimalizovány pro aplikace, které získávají kompletní entity, které odpovídají požadovaným kritériím. Naopak sloupcový přístup k ukládání dat je výhodný pro zpracování Big Data, protože jsou optimalizovány dotazy, které jsou formulovány pro získání jednoho sloupce nebo skupiny sdružených sloupců mnoha záznamů, spíše než dotazy, které získávají všechna uložená data o jedné entitě. Výhody SQL Data Warehouse jsou patrné v situacích, kdy jsou data o velikosti miliard záznamů uložené v tabulkách, které je třeba optimalizovat a rozdělit do oddílů.



Obrázek 3.1: Koncept architektury MPP. Zdroj: [17].

Technologie MMP využívá architekturu s výpočetními uzly. Aplikace jsou připojeny k centrálnímu uzlu a vytvářejí dotazy v jazyce T-SQL. Centrální uzel využívá MPP engine, který dotaz rozdělí na dílčí úkony, které provádějí jednotlivé výpočetní uzly. Výpočetní uzly využívají DMS (Database Migration Servise) pro sdílení potřebných dat mezi sebou. Data ve výpočetních uzlech jsou často uložena v nestrukturovaných úložištích Azure Blob Storage(viz kapitola 3.4.2.1) kvůli nízké ceně, odkud jsou nahrávána do relační podoby pomocí technologie PolyBase, která je vytvořena pro tento účel. Výsledky jednotlivých operací výpočetních uzlů jsou vráceny do centrálního uzlu, odkud jsou získané výsledky předány aplikaci.

3.4 Úložiště nestrukturovaných a semi-strukturovaných dat

Nerelační databáze nevyužívá schéma tabulek, které jsou propojeny relacemi, jedná se tedy o "bez schématické úložiště". Namísto ukládání dat do ER Modelu je forma uložení volena na míru ukládaným nestrukturovaným datům. Existuje několik základních přístupů, které se používají k uchovávání Big Data na MS Azure v nerelačním formátu.

Pro procházení dat v nestrukturovaných úložištích je vhodný nástroj Microsoft Azure Storage Explorer. Ten umožňuje dotazování a filtrování tabulek typu klíč-hodnota, prozkoumávání souborové struktury v Azure Blob Storage a Azure Data Lake Store a mnoho dalších funkcí. Dále jsou přiblíženy jednotlivé způsoby ukládání nestrukturovaných dat v nerelačních úložištích.

3.4.1 Databáze klíč-hodnota(Key-Value)

V úložišti typu Key-Value se ukládají serializované záznamy spárované s jejich unikátním klíčem. Úložiště, do kterého se ukládají páry klíč-hodnota, připomíná tabulku. Algoritmus, kterým je volen klíč, se nazývá hashování a velmi ovlivní výkon úložiště. Tento typ je vhodný pro ukládání a vyhledávání hodnot podle klíče. Naopak není optimalizovaný pro vyhledávání podle obsahu záznamu. Je tedy důležité vhodně zvolit algoritmus hashování klíče. Příklad služby na MS Azure je Azure Table Storage která umožňuje ukládání záznamů do tabulky, které mají podobný formát jako v Azure SQL databázi. Na rozdíl od ní však neumožňuje vytváření relací mezi jednotlivými tabulkami.

Struktura úložiště V každém účtu Azure Table Storage je možné vytvořit mnoho tabulek. Každý řádek v tabulce představuje uložený unikátní záznam, který má až 255 datových políček (sloupců) a maximální velikost 1 MB.

Z 255 možných datových polí jsou tři údaje, RowKey, PartitionKey a Timestamp vytvořeny již při vytváření záznamu. RowKey je primární a unikátní index používaný pro identifikaci záznamu, PartitionKey definuje do které části tabulky se záznam uloží, což optimalizuje ukládání z hlediska urychlení dotazování úložiště, a Timestamp je časová značka, které se používá pro určení časového okamžiku vzniku záznamu a filtrování záznamů.

Obrovskou nevýhodou Azure Table Storage je plochá struktura tabulek, což znamená, že není možné vytvářet relace mezi tabulkami a využívat tak potenciál ER modelu. Také není vhodné do tabulek Azure Table Storage ukládat nestandardní datové typy, například pokud je potřeba uložit do jednoho pole řádku seznam, je nutné ho ponechat v

serializovaném JSON/XML formátu, podle kterého se však nedá vyhledávat a filtrovat. Dotazování ve tvaru "Najdi prvek, který má v tomto seznamu X." tedy není možné nebo je velmi špatně optimalizované. Spojovat data a vykonávat agregační výpočty je v tomto úložišti nevhodné a také se obtížně realizují dotazy spojující data z více tabulek.

Škálovatelnosti úložiště a dotazování dat Objemový tok Azure Table Storage je omezen na 20 000 požadavků za sekundu [11] při velikosti záznamu 1 KB. Další omezení je 2 000 požadavků za sekundu z jedné partition. Maximální objem úložiště je 500 TB. V porovnání Azure Table Storage umožňuje lepší škálování a nižší cenu než relační SQL databáze, ale není na ní možné vznášet složité dotazy spojující více tabulek.

3.4.2 Objektová úložiště

Objektová datová úložiště jsou optimalizována pro ukládání a přechovávání velkých binárních objektů, které jsou známé pod označením BLOB (Binary Large OBject). Tyto objekty mohou reprezentovat jakékoliv digitální objekty, dlouhé záznamy textu, média nebo celé obrazy disků.

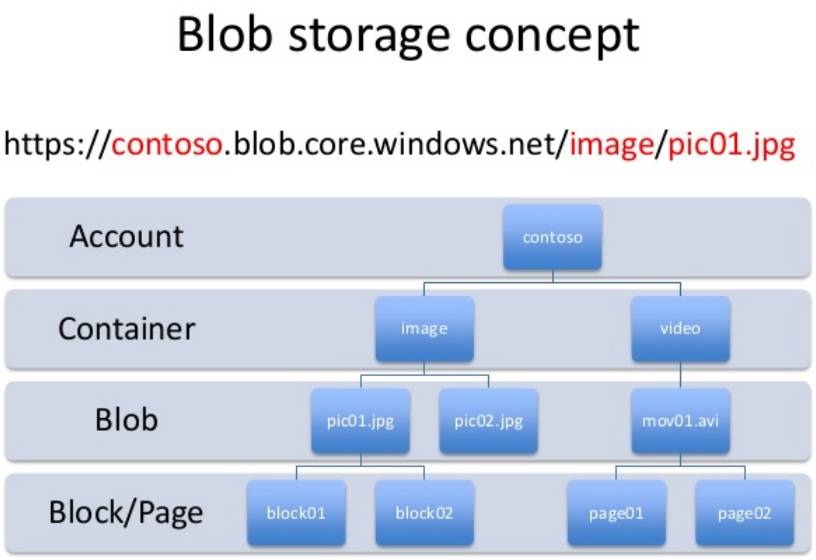
Struktura úložiště Souborové binární bloky jsou tříděny do adresářové struktury, která slouží k hledání dat konkrétních záznamů. Nevyhledává se tedy podle datového obsahu, ale pouze podle umístění v adresářové struktuře. Některé implementace tohoto úložiště používají zrcadlení částí souborů na více serverech, které umožňují rychlejší a paralelní zpracovávání dat a přístup k zápisu a čtení najednou. Jako speciální typ blobového úložiště může být považováno klasické sdílení souborů v síťové složce. Na Azure se nejčastěji pro tyto úložiště využívají služby Azure Blob Storage, Azure Data Lake Store a Azure File Storage.

3.4.2.1 Azure Blob Storage

V každém účtu Azure Blob Storage existuje více kontejnerů. Kontejnery mají uvnitř plochou strukturu a obsahují soubory s daty. Blobové úložiště je ideální pro ukládání souborů na více míst pro distribuci obsahu k analýze, streamování videa a audia, záznam logů jiných aplikací nebo ukládání obrazů stavu jiných služeb za účelem zálohy a obnovení služby.

Existují tři typy ukládání do blob úložiště: Block, Append a Page [22]. V Block Blobu jsou ukládána textové nebo binární data až do velikosti 4 TB a tyto bloky dat jsou samostatně zpracovatelné. Tento způsob je velmi vhodný pro ukládání médií nebo

jiných, nedělitelných objektů. Append Blob je podobně jako Block složen z bloků dat a je optimalizován pro přidávání jednotlivých záznamů do bloku dat, například přidávání vstupů do logu na konec souboru. Page Blob je kolekcí 512 bytových stránek, které umožňují přístup k jednotlivým stránkám a jsou vhodné pro aplikace, kde je třeba více indexovat obsah, protože se k němu přistupuje kvůli čtení nebo editaci.



Obrázek 3.2: Struktura Azure Blob Storage. Ukázka rozmístění dat do adresářové struktury. Contoso je fiktivní firma vytvořená Microsoftem pro ilustrativní účely. Zdroj: [6].

Škálovatelnosti úložiště a dotazování dat Azure Blob Storage má omezení maximálního objemu uložených dat 500 TB a omezení 20 000 požadavků za sekundu, podobně jako u úložiště Azure Table Storage. Dotazovaní dat v Azure Blob Storage je jeho slabé místo, protože prohledávání dat podle obsahu bez kompletního zpracování blobu není možné. Při prohledávání dat je možné se orientovat podle názvu souboru, kam je možné zahrnout nějaké prvky obsahu uloženého objektu. Některá řešení si dokonce k blobovému úložišti vytváří paralelní tabulku, kam se ukládají vyhledávací indexy pro každý uložený záznam a výběr patřičného blobu pro analýzu probíhá tam.

3.4.2.2 Azure Data Lake Store

Azure Data Lake Store (ADLS) [26] je vysoce škálovatelné úložiště určené pro analýzu velkých objemů dat. ADLS umožňuje ukládání dat jakékoliv velikosti, jakéhokoliv typu a použití neomezené rychlosti zápisu.

Je možné uložit data v podstatě libovolného typu, protože ADLS nevyžaduje vytvoření explicitního schématu při ukládání. Takže si každá analytická úloha nebo nástroj může ukládat data ve formátu, který je pro konkrétní úlohu nejefektivnější. ADLS tedy může ukládat strukturovaná, semi-strukturovaná i nestrukturovaná data. Základní způsob ukládání dat do ADLS je využívání složek a souborů, mezi které je obsah tříděn. Například zprávy, uvažovány v této práci(viz kapitola 4.1), jsou v textovém formátu. Je tedy možné definovat adresářovou strukturu, do které budou uložené textové soubory obsahující například všechny zprávy z dané hodiny.

Při využití relačního řešení, například ukládání dat v SQL databázi, je možné využít ADLS jako úložiště dat. Z něj se použije framework Sqoop nebo PolyBase pro nahrání záznamů z nestrukturovaného úložiště do relačního schématu. V tomto řešení je SQL důležitá pro udržení konsistence a relací a ADLS je daleko lépe optimalizováno pro analytické úlohy a je levnější.

Úložiště ADLS je vhodné pro analýzu velkých objemů dat, protože rozděluje data na části, které ukládá na různých serverech. To odemyká cestu k paralelnímu zpracovávání dat a výraznému zrychlení. Zároveň se díky své flexibilitě využívá jako výchozí datové úložiště výpočetních služeb.

Škálovatelnosti úložiště a dotazování dat Microsoft uvádí, že ADLS nemá omezení ve velikosti uložených dat nebo rychlosti přistupování. Dotazovatelnost dat závisí na typu uložených dat. Při analýze dat se používají frameworky, které si data naindexují samy a ADLS využívají jako datovou bázi. Samotný přístup k datům je podobný jako v Azure Blob Storage. Musí být načteny všechny části obsahující požadovaná data a ty pak zpracovány.

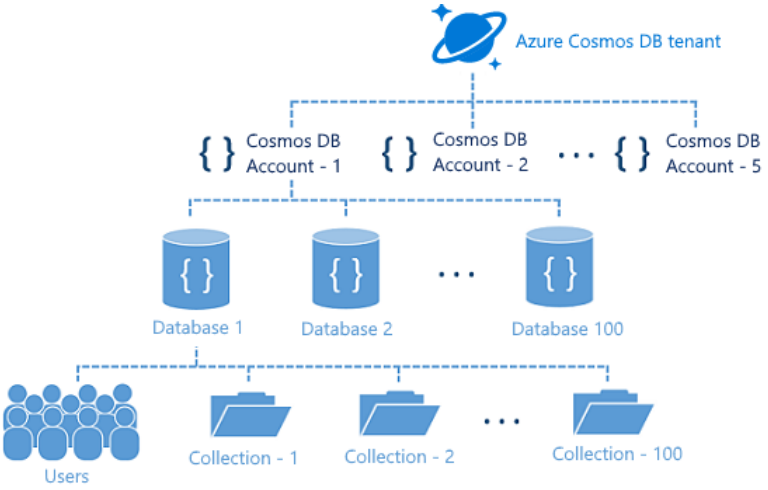
Porovnání Azure Data Lake Store s Azure Blob Storage Na první pohled se může zdát, že se jedná o podobný produkt. V obou se ukládají data do souborového systému. ADLS obsahuje adresářovou strukturu, ve kterých jsou uložena data ve formě souborů. Azure Blob Storage obsahuje kontejnery, které v sobě obsahují sady blobových souborů. Oba systémy podporují distribuování souborů na různé servery kvůli umožnění paralelních výpočtů.

ADLS však má výhodu [13] v tom, že umožňuje v podstatě neomezenou velikost ukládaných dat a velikosti datových toků. Azure Blob Storage je limitován maximální velikost 500 TB. Významnější je však omezení 20 000 požadavků za sekundu, což může podstatně zpomalit rychlost čtení při analýzách. Azure Blob Storage je tedy ze své podstaty vhodný spíše pro jiné úkoly, než který je zamýšlen v této práci. Příkladem může být ukládání logů využitých služeb.

3.4.3 Dokumentová úložiště

Dokumentová úložiště se zaměřují na ukládání textových dokumentů. Každý dokument má svůj unikátní identifikátor. Typickým formátem pro ukládání informací v textovém formátu jsou JSON a XML. Pro každý uložený záznam se vytváří jeden dokument. Protože není omezena hloubka struktury, dokumentové úložiště je vhodné pro ukládání rozsáhlých hlubokých JSON a XML reprezentací objektů. Unikátní klíč, který je určený pro přístup k dokumentu, se z důvodu snadnějšího vyhledávání často skládá z důležitých údajů v záznamu. Pro jeho generování se používají hashovací algoritmy stejně jako u Azure Table Storage (viz kapitola 3.4.1). Běžnou službou využívající dokumentové úložiště je Azure Cosmos DB.

Struktura úložiště Pro každý účet vytvořený ve službě Azure Cosmos DB je možné vytvořit více databází. Každá databáze v sobě obsahuje více datových kontejnerů. V každém kontejneru je třeba specifikovat zda se ukládají kolekce, dokumenty, grafy nebo tabulky a také je zvolen požadovaný průtok dat a maximální kapacita. Schéma Azure CosmosDB je ukázáno na obrázku 3.3 Kromě uložených záznamů obsahují kontejnery uložené procedury určené pro automatické akce prováděné při ukládání souborů. Dokumentové úložiště je příkladem semi-strukturovaného úložiště, protože neexistuje vztah mezi jednotlivými uloženými záznamy, které ale mohou mít hlubší strukturu a jsou od sebe jednoznačně odděleny.



Obrázek 3.3: Struktura Azure Cosmos DB služby. Zdroj: [21]

Škálovatelnosti úložiště a dotazování dat Stejně jako u ADLS, Microsoft uvádí, že Azure Cosmos DB nemá omezení velikosti uložených dat a rychlosti přístupu k datům.

Tyto vlastnosti je možné měnit, například zvýšit v aktivních hodinách dne. Pro charakterizaci rychlosti přístupu se používá jednotka RU.

Request Unit (RU) RU je normalizovaná jednotka datového přenosu, která odpovídá je zápisu/čtení 1 kB zprávy. V Azure Cosmos DB se využívá jednotka RU/s, která charakterizuje požadovaný výkon, který od služby bude vyžadován. Před založením služby je tedy vhodné rozmyslet jak velké jsou zprávy a kolik je jich za sekundu očekáváno.

Dotazování dat je silnou stránkou Azure Cosmos DB. Využívá se známá syntaxe SQL a je možné se pomocí tečkové notace dostat až do hloubky záznamu. Jsou podporovány operace shlukování GROUP BY a filtrování FILTER, které jsou velmi důležité při transformacích a třídění dat. Návratová hodnota těchto dotazů je formát JSON, který snadno umožní další zpracování výsledků.

3.4.4 Grafová úložiště

Odlišný přístup k ukládání dat přináší grafové úložiště [5]. Grafové úložiště se skládá z vrcholů propojených hranami. Vrcholy obsahují data a hrany reprezentují relace mezi vrcholy. Takovéto úložiště umožňuje reprezentovat velmi složitou provázanou strukturu dat, kterou by bylo obtížné vyjádřit tabulkou nebo v JSON. Grafové úložiště umožňuje provádět efektivní opakované prohledávání podle referencí. Grafové úložiště je implementováno například ve službách Azure Cosmos DB Graph API a Neo4j. V této práci však není třeba tyto služby dále popisovat, protože použitá data nejsou vhodná pro grafové úložiště.

3.5 Výběr úložiště

Před vyhodnocením, která varianta úložiště je vhodná pro účely této práce, je třeba uvést porovnání jejich cen.

3.5.1 Analýza podle ceny

U vybraných komponent je obtížné porovnávat jejich ceny, protože používají odlišné platební modely. U služeb se obvykle platí za objem přechovávaných dat a za přístup k nim. Azure Table Storage, Azure Blob Storage a Azure Data Lake Store při cenění přístupu k datům používají čtení nebo zápis 10 000 normalizovaných záznamů. Azure

SQL Database a Cosmos DB používají jednotky DTU a RU/s, které jsou v poznámkách k tabulce 3.1 orientačně přepočítány na jednotky používané v Azure Data Lake Store.

Jako referenční cena objemu dat je vybrána cena za uchovávání 1GB za měsíc. Jako referenční cena dotazování dat byla vybrána cena za čtení nebo zápis 10 000 normalizovaných záznamu.

Tabulka 3.1: Tabulka cen úložišť na MS Azure.

	Cena za 1 GB	Cena za čtení
Azure Storage (Blob)	$0.01	$0.01
Azure Cosmos DB (Document)	$0.25	viz Poznámky
Azure Data Lake Store (Blob)	$0.039	$0.004
Azure Storage (Table)	$0.07	$0.00036
Azure SQL Database (Relational)	$0.12	$0.0005 viz Poznámky

Poznámky k tabulce 3.1:

- Vybrané ceny jsou platné pro Západní Evropu, jsou v USD a zjišťované byly 2.5.2018.
- Cena pro Azure Storage Blob je pro lokální datovou redundanci. Pokud by bylo nutné mít zaručenou vyšší míru dostupnosti, je cena vyšší. V této práci však zdaleka postačí tato verze. U Azure Blob Storage existují různé cenové modely, které se liší poměrem ceny za skladování dat a ceny za přístup k datům: Hot, Cool a Archive. U Hot je vysoká cena za skladování dat a nízká cena za přístup k uloženým datům. U Archive je nízká cena za skladování dat ale vyšší cena za každé jejich přečtení (neplánuje se k datům často přistupovat). Pro analýzu vlastností je zvolena varianta Cool, která je doporučena pro dlouhodobé ukládání nezpracovaných dat, které jsou určeny pro budoucí analýzu.
- Azure Cosmos DB má odlišný platební model. Platí se zde za místo na SSD a za RU/s. Nelze mít méně než 400 RU/s. Za 400 RU/s je cena $23.35 za měsíc. Nepředpokládá se, že bude zapisováno více než 400 Kb/s v této práci, minimální objem Azure Cosmos DB by byl pravděpodobně dostačující.
- Cena za operace v Azure SQL Databázi je definována pomocí jednotky DTU. Za použití online Microsoft kalkulačky pro DTU vyšlo, že provedení 10 000 operací za sekundu po dobu jedné hodiny stojí 2$, což je, přepočítáno na sekundy 0.0005$.

3.5.2 Vyhodnocení a analýza možných scénářů využití

Pro účely této práce je využita kombinace relačního i nerelačního úložiště. Využije se tak kombinace flexibility a ceny Azure Data Lake Store a integrity dat Azure SQL Database. Data jsou při příchodu do systému ukládána do ADLS, kde jsou roztříděna podle času a zařízení, ze kterého přicházejí. Odtud jsou za pomocí služby Azure Data Factory transformována a ukládána do relačního úložiště Azure SQL Database, odkud si data dotazují analytické a vizualizační služby. Tento systém je ilustrován v obrázku 2.4.

Jako nerelační úložiště je vybráno Azure Data Lake Store kvůli jeho flexibilitě a ceně. Škálování ADLS zaručí, že při neočekávaném navýšení objemu přijímaných dat nedojde k selhání úložiště. Flexibilita úložiště umožňuje ukládání širokého spektra dat z analytických nástrojů, například metadata, výsledky, metriky, tabulky, grafy nebo objekty jednotlivých verzí klasifikátorů strojového učení. Data jsou převáděna do relačního úložiště Azure SQL Database, které je díky konzistenci a integritě uložených dat výhodnou platformou pro následnou analýzu a vizualizaci dat.

Kapitola 4

Vytvoření datového úložiště

V předchozí kapitole jsou shrnuty různé varianty datových úložišť, z nichž byly vybrány ty, které jsou připravené pro nástroje datové analýzy a strojového učení. V této kapitole je popsán sběr dat do nestrukturovaného úložiště a transformace dat do relační databáze pomocí Azure Data Factory.

4.1 Zdroje Dat

Data používané v této práci se dají rozdělit do tří základních kategorií podle jejich původu a významu na servisní, jízdní a statická. Pro každou kategorii je připravena složka v Azure Data Lake Store, kde jsou zprávy shromažďovány a odkud jsou dále zpracovávány.

4.1.1 Servisní Data

Servisní data jsou odesílána z mobilní aplikace při návštěvě servisu. Obsahují informace o vozidle a prováděné proceduře nebo diagnostická data. Ve výpisu 4.1 je ukázána servisní zpráva, která se odešle když technik zvolí akci 'EOBD Read' a přečte chybová hlášení vozidla. Zpráva obsahuje unikátní identifikátor, typ akce, data k vybrané akci a informaci, zda akce proběhla úspěšně či nikoliv. Časová informace ve zprávě chybí, je však přidána ke zprávě při průchodu IoT Hubem.

```
{
    "_id" : ObjectId("5afa43a01301820001b7edd4"),
    "name" : "eobd_read",
    "info" : {
```

```
        "mil" : "Off",
        "dtcs" : [
            {
                "state" : "current",
                "name" : "P100",
                "desc" : "can1 fault data"
            },
            {
                "state" : "current",
                "name" : "P101",
                "desc" : "can1 fault data"
            }
        ],
        "protocol" : "ISO15765",
        "user_id" : "a1b2c3d4e5",
        "vin" : "YV1MK754172017090",
        "dtc_count" : 2,
        "success" : "true"
    }
}
```

Výpis 4.1: Ukázka zprávy, která se odešle při přečtení chybových hlášení ve vozidle při návštěvě servisu a připojení ZF Smart Device do vozidla.

4.1.2 Jízdní Data

Jízdní data jsou odesílána z Vivaldi Boxu, který je nainstalovaný ve vozidle a snímá široké spektrum signálů. Protože jízdní data využívá mnoho jiných aplikací v OPENMATICS, existuje společná brána (IoT Hub) pro zprávy z Vivaldi Boxů, která data třídí mezi systémy jednotlivých projektů pro různé zákazníky.

V této práci je vybrán vzorek vozidel, pro který jsou vybrány zprávy obsahující GPS data, data z vybraných senzorů a chybové zprávy DTC (Diagnostic Trouble Code). Ukázky těchto zpráv jsou ve výpisech 4.2, 4.3 a 4.4.

```
{
    "DeviceId":"973093000670",
    "Type":"Data",
    "Timestamp":"2018-06-12t01:22:00.0000000z",
    "Payload":{
        "battery_voltage":"12.71",
        "accumulated_mileage":"9134.93",
        "mileage id":"654",
        "eventtime":"2018-06-12t01:21:31.0000000z",
        "ts_dt":"2018-06-12T00:00:00.0000000",
        "ts_1min":"2018-06-12T01:21:00.0000000",
        "ts_10min":"2018-06-12T01:20:00.0000000",
        "ts_1hr":"2018-06-12T01:00:00.0000000",
        "ts_3hr":"2018-06-12T00:00:00.0000000",
```

```
        "ts_6hr":"2018-06-12T00:00:00.0000000",
        "Fleet":"ZF - Turkey",
        "LicencePlate":"34RT5848"
    },
    "EventProcessedUtcTime":"2018-06-12T01:22:03.5728422Z",
    "PartitionId":0,
    "EventEnqueuedUtcTime":"2018-06-12T01:22:03.5380000Z"
}
```

Výpis 4.2: Ukázka zprávy, obsahující signál o ujetých kilometrech a stavu nabití baterie. Každá zpráva může mít jinou sadu obsažených dat, protože různé senzory mají jinou periodu měření.

```
{
    "DeviceId":"973093000670",
    "Type":"Gps",
    "Timestamp":"2018-06-12t03:58:12.0000000z",
    "Payload":{
        "eventtime":"2018-06-12t03:58:08.0000000z",
        "trip_sn":655,
        "lon":29.121266666666664,
        "lat":40.935411666666667,
        "accuracy":5,
        "direct":320,
        "mode":1,
        "speed":31,
        "alt":10,
        "carspeed":46,
        "ts_dt":"2018-06-12T00:00:00.0000000",
        "ts_1min":"2018-06-12T03:58:00.0000000",
        "ts_10min":"2018-06-12T03:50:00.0000000",
        "ts_1hr":"2018-06-12T03:00:00.0000000",
        "ts_3hr":"2018-06-12T03:00:00.0000000",
        "ts_6hr":"2018-06-12T03:00:00.0000000",
        "Fleet":"ZF - Turkey",
        "LicencePlate":"34RT5848"
    },
    "EventProcessedUtcTime":"2018-06-12T03:58:14.7351970Z",
    "PartitionId":0,
    "EventEnqueuedUtcTime":"2018-06-12T03:58:14.5620000Z"
}
```

Výpis 4.3: Ukázka zprávy Data, která představuje měření GPS modulu. Zpráva se odešle každé tři sekundy.

```
{
    "DeviceId":"973093000524",
    "Type":"Dtc",
    "Timestamp":"2018-06-01t13:53:36.0000000z",
    "Payload":{
        "data":[
```

```
            {
                "faultdes":"o2 sensor heater circuit bank 2 sensor 2",
                "faultid":"00000161",
                "sysid":"ffffffff",
                "faultvalue":"p0161",
                "faultstatus":2
            },
            {
                "faultdes":"o2 sensor heater circuit bank 1 sensor 2",
                "faultid":"00000141",
                "sysid":"ffffffff",
                "faultvalue":"p0141",
                "faultstatus":2
            },
            {
                "faultdes":"o2 sensor circuit no activity detected bank 2 sensor 2",
                "faultid":"00000160",
                "sysid":"ffffffff",
                "faultvalue":"p0160",
                "faultstatus":2
            },
            {
                "faultdes":"o2 sensor circuit no activity detected bank 1 sensor 2",
                "faultid":"00000140",
                "sysid":"ffffffff",
                "faultvalue":"p0140",
                "faultstatus":2
            }
        ],
        "eventtime":"2018-06-01t13:53:36.0000000z",
        "ts_dt":"2018-06-01T00:00:00.0000000",
        "ts_1min":"2018-06-01T13:53:00.0000000",
        "ts_10min":"2018-06-01T13:50:00.0000000",
        "ts_1hr":"2018-06-01T13:00:00.0000000",
        "ts_3hr":"2018-06-01T12:00:00.0000000",
        "ts_6hr":"2018-06-01T12:00:00.0000000",
        "Fleet":"ZF Aftermarket",
        "LicencePlate":"524"
    },
    "EventProcessedUtcTime":"2018-06-01T13:53:39.5643811Z",
    "PartitionId":0,
    "EventEnqueuedUtcTime":"2018-06-01T13:53:39.5080000Z"
}
```


Výpis 4.4: Ukázka zprávy, která zaznamenává varování a poruchy součástek vozidla.

Existují i další typy zpráv, které se posílají při jízdě a nejsou zde z důvodu stručnosti uvedeny, například zprávy o začátcích a koncích jízdy, závadách v průběhu jízdy, přihlášení nového vozidla po registraci a další.

4.1.3 Statická Data

Statická data nejsou přijímána ve formě zpráv. Jedná se o informace o vozidlech, servisech nebo osobách, které jsou zadávány do firemního portálu nebo se získávají jednorázově při instalaci jednoho ze zařízení do vozidla. Tato data jsou uložena ve firemní SQL databázi, odkud jsou pro účely této práce nakopírována do příslušné tabulky relačního úložiště(viz kapitola 4.3.1) pro daný vzorek vozidel.

4.2 Shromáždění dat v nerelačním úložišti

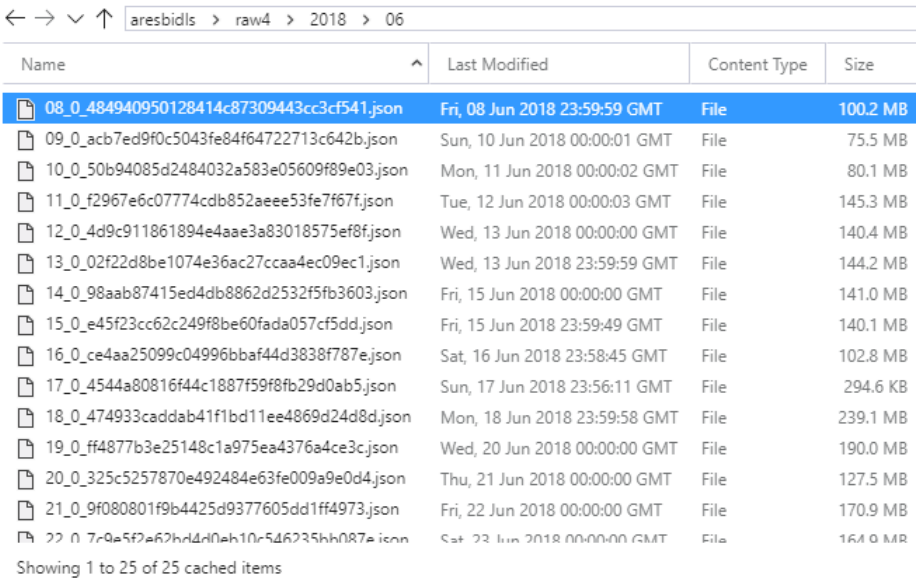
Jízdní a servisní data jsou shromažďována v komponentě Event Hub, odkud jsou pomocí komponenty Stream Analytics Job ukládána do blobového úložiště v Azure Data Lake Store. Tento postup je popsán a ukázán v kapitole 2.3.6.

Event Hub je komponenta pro streamování dat [23], která je schopná přijímat velké množství zpráv, událostí a telemetrických informací. Data mohou být po krátkou dobu uchovávána v Event Hubu, odkud jsou transformována, tříděna a ukládána do permanentních úložišť.

Stream Analytics Job je engine pro zpracovávání událostí, který umožňuje zpracování dat streamových z mnoha zdrojů, například zařízení, senzorů, aplikací, Iot Hubu, Event Hubu a dalších. Uvnitř Stream Analytics Jobu se používá jazyk podobný SQL, který umožňuje základní analýzu dat a třídění dat. Jako výstup zde může být nakonfigurován další Event Hub, SQL databáze, nerelační úložiště a jiné. V navrhované aplikaci jsou zprávy krátkodobě uložené v Event Hubu transformovány do velkých blobových souborů v Azure Data Lake Store.

Složky vytvořené pro servisní a jízdní záznamy obsahují zprávy roztříděné do souborů podle času přijetí, jak je ukázáno na obrázku 4.1. V těchto souborech jsou zaznamenány série jízdních zpráv v textovém formátu. K prohlížení dat uloženým v Azure Data Lake Store, Azure Table Storage nebo Azure Blob Storage je využit počítačový program Microsoft Azure Storage Explorer.

Jízdní data do ADLS proudí za pomocí Stream Analytics Job z Event Hubu s názvem 'raw4' , proto jsou ukládána do složky s tímto názvem. Uvnitř je adresářová struktura ve tvaru /rok/měsíc/, ve které jsou uloženy jednotlivé soubory. Na začátku názvu každého souboru je vidět, ze kterého dne jsou v něm obsaženy záznamy. V podobné struktuře jsou ukládána servisní data z ZF Smart Device.



Obrázek 4.1: Strukura v nerelačním blobovém úložišti Komponenta typu Azure Data Lake Store s názvem 'aresbidls'.

4.3 Příprava modelu v SQL databázi

V této části je navržen relační model určený pro uložení dat, které jsou použity v agregačních výpočtech, úlohách strojového učení a vizualizacích. Model obsahuje tabulky pro statická data, jízdní data, servisní data a další tabulky vytvořené pro agregační analytické výpočty a využívání klasifikátoru strojového učení.

Uvedený model není zatím kompletní, protože v době psaní této práce nejsou známy všechny typy servisních zpráv kvůli tomu, že zatím nebyl definován kompletní seznam servisních úkonů podporovaných zařízením ZF Smart Device. Model je tedy obecný a vytvořen s ohledem na reálný příklad využití tohoto analytického systému, predikce výměny brzdových destiček, který je popsán v kapitole 6.1.

4.3.1 Tabulka Ref_Vehicle

V tabulce 4.1 jsou obsaženy údaje o vozidlech. Data jsou získávána z firemní databáze vozidel a jsou periodicky obnovována. Zároveň tato tabulka obsahuje datové sloupce pro agregační a statistická data, které jsou získávána při běhu analytických nástrojů, například počítání událostí, počtu zpráv, aktivní doby vozidel a podobně.

Název pole	Význam pole
deviceId	Identifikátor Vivaldi Boxu, který je nainstalovaný ve vozidle.
vehicleType	Typ vozidla.
mileageTotal	Jaká je celková ujetá vzdálenost od počátku sběru dat Vivaldi Boxem.
status	Zda bylo vozidlo aktivní za poslední jeden den.
fuelType	Typ pohonu vozidla - plyn, elektřina, benzín nebo nafta.
mileageSinceService	Jaké je ujetá vzdálenost od poslední návštěvy v servise.
recordLastModification	Kdy byl tento záznam v tabulce naposledy upraven.

Tabulka 4.1: Popis datových polí v tabulce Ref_Vehicle

Název pole	Význam pole
workshopId	Identifikace servisu
workshopName	Označení servisu
latitude	Zeměpisná šířka servisu.
longitude	Zeměpisná délka servisu.
phone	Kontakt na provozovatele servisu.
personName	Jméno mechanika provádějící opravy.

Tabulka 4.2: Popis datových polí v tabulce Ref_Workshop.

4.3.2 Tabulka Ref_Workshop

V tabulce 4.2 jsou uvedeny údaje o servisech, které odesílají data se servisními úpravami provedených na vozidlech. Jsou zde informace o majiteli, poloze a kontaktní údaje. Data jsou získána z dokumentace k ZF Smart Service.

4.3.3 Tabulka Data

V tabulce 4.3 jsou obsažena data ze zpráv s GPS, Daty (diagnostické signály) a DTC (poruchy), ukázaných ve výpisech 4.2, 4.3 a 4.4. Tabulka obsahuje pole pro uchování informací všech tří typů zpráv. Protože má zpráva typu Data mnoho datových položek, které nejsou zatím potřeba, jsou ponechány pouze položky 'battery_voltage' a 'accumulated_mileage'. Ostatní položky zprávy mohou být přidány později. Sloupec 'velocityDelta' je dopočítáván v analytickém nástroji. Položky DTC zprávy faultId, sysId, faultValue, faultStatus a faultDesc jsou popsány například ve článku [2].

4.3.4 Tabulka Service_Brake_Reset

V tabulce 4.4 jsou ukládány záznamy o návštěvách servisu a vykonání akce výměny brzd. Každý datový záznam v této tabulce je spojen se záznamem v Ref_Vehicle a Ref_Workshop.

Název pole	Význam pole
deviceId	Označení vozidla, ke kterému přísluší tento datový záznam.
timestamp	Časová značka záznamu.
latitude	Zeměpisná šířka záznamu.
longitude	Zeměpisná délka záznamu.
velocity	Aktuální rychlost.
faultId	DTC - Popis poruchy ve vozidle.
sysId	
faultValue	
faultStatus	
faultDes	
battery_voltage	Hodnota napětí autobaterie.
accumulated_mileage	Celková ujetá vzdálenost od instalace Vivaldi Boxu do vozidla.
velocityDelta	Změna rychlosti jízdy oproti předchozímu záznamu vozidla.

Tabulka 4.3: Popis datových polí v tabulce Data.

Název pole	Význam pole
deviceId	Označení vozidla, na kterém byla akce provedena.
timestamp	Časová značka záznamu.
workshop	Označení Workshopu, ve které by akce provedena.
typeBefore	Typ brzdy, které byly ve vozidle před návštěvou servisu.
typeAfter	Typ brzdy, které byly nainstalovány při této akci.
success	Indikátor, že byla akce úspěšně provedena.

Tabulka 4.4: Popis datových polí v tabulce Service_Brake_Reset.

Název pole	Význam pole
deviceId	Označení vozidla, ke kterému přísluší tento datový záznam.
timestamp	Časová značka záznamu.
latitude	Zeměpisná šířka záznamu.
longitude	Zeměpisná délka záznamu.
eventType	Označení jízdní události.
eventValue	Hodnota veličiny, kterán je předmětem jízdní události.

Tabulka 4.5: Popis datových polí v tabulce Events, která je určena pro záznam jízdních událostí.

4.3.5 Tabulka Events

V tabulce 4.5 jsou zaznamenány jízdní události, které jsou dopočítávané při běhu analytických nástrojů. Zatím je definován jeden typ události ve třech variantách, 'dccel30-40','dccel40-50' a 'dccel50-90', které vyjadřují prudké zpomalení vozidla o příslušnou hodnotu km/h.

Název pole	Význam pole
Id	Unikátní klíč
deviceId	Označení vozidla
timestamp_created	Časová značka vytvoření záznamu.
timestamp_processed	Časová značka vyhodnocení záznamu.
feature1 Vstupní data do modelu.	
...	
feature10	
result Výstupní data z modelu.	

TABULKA 4.6: Popis datových polí v tabulce Ml_Brake_Service, která se používá pro komunikaci s webovou službou.

4.3.6 Tabulka Ml_Brake_Service

V tabulce 4.6 jsou uložena data využívaná při používání modelů pro vytváření predikcí. Jsou zde uložena data pro vstup do modelu a připravena pole, která se zapíšou při vyhodnocení požadavku. Mechanismus správy této tabulky je popsán v dalších kapitolách. Tato tabulka bude používána pouze v ilustrativním příkladu této práce. Sloupce 'Timestamp_processed' a 'result' jsou vyplněny při volání webové služby s prediktivním modelem, která využívá ostatní vyplněná pole.

4.4 Převod dat do SQL databáze

Po přípravě dat v nestrukturovaném úložišti a vytvoření modelu v SQL databázi je možné nadefinovat akci, která bude kopírovat data podle daného mapování sloupců. Pro tento účel je vhodná služba Azure Data Factory.

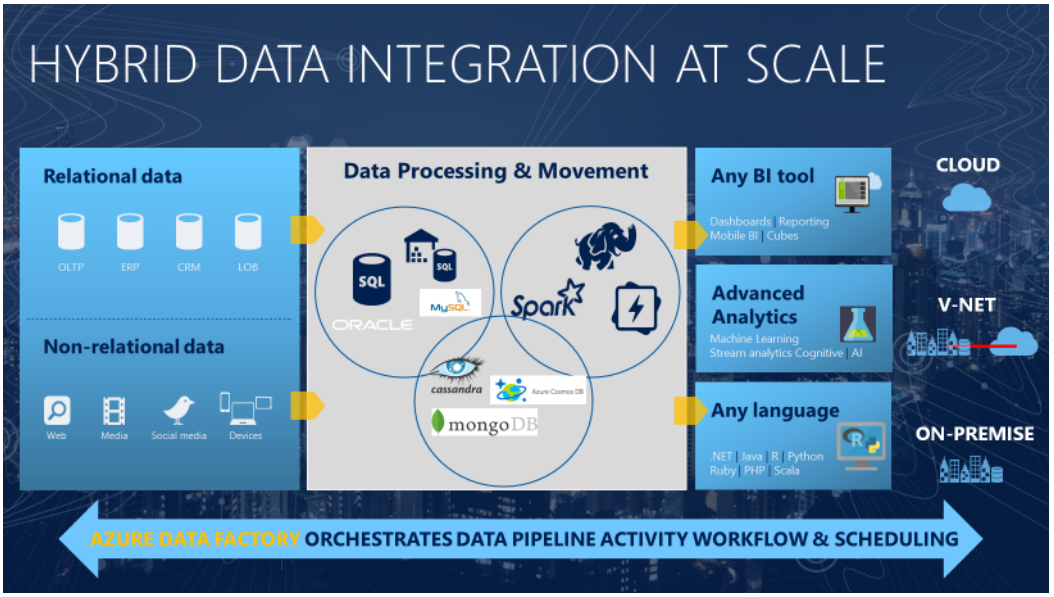
Azure Data Factory je cloudová služba, která je určená pro řízení datových procesů v Big Data projektech. Obsahuje v sobě schémata, kde jsou popsány datové a výpočetní komponenty a spojení (pipeline) mezi nimi, které mohou být spouštěny periodicky nebo při definované události.

Tato schémata obsahují nástroje pro všechny části ELT (Extract-Load-Transform) architektury, ukázané na obrázku 4.2. Vlevo jsou relační a nerelační zdroje dat (Extract). Uprostřed jsou datová úložiště optimalizované pro datovou analýzu (Load) - Azure SQL Database, Azure Data Lake, Mongo/Document DB. Vpravo jsou nástroje pro datovou analýzu a reporting (Transform).

Jednoduché schéma, které je použito pro kopírování dat z Azure Data Lake Store do Azure SQL Database, je ukázáno na obrázku 4.3. Ve schématu jsou dva druhy akcí. První

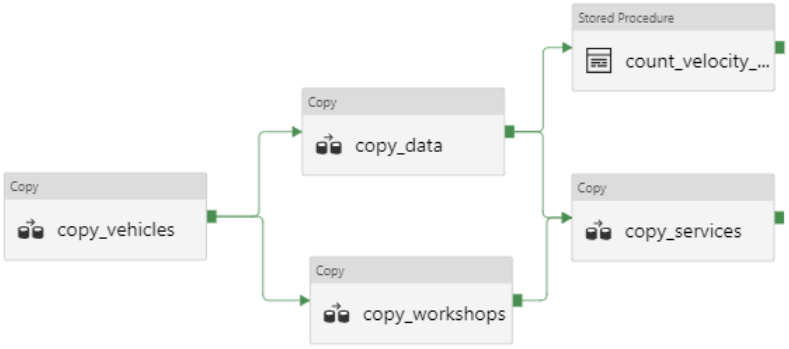
je 'Copy Data', která kopíruje data ze zdrojového do cílového úložiště. V této akci je třeba nadefinovat datová pole a jejich typy ve zdroji i cíli a vytvořit mapování mezi nimi. Při kopírování je možné přidávat k záznamům dynamické datové položky, jako například informace o běhu pipeline, při kterém byla zpráva zpracována nebo transformace obsahu zprávy libovolnou funkcí.

Druhým typem je Stored Procedure. Jedná se o akci, která je definována v Azure SQL Database, spouští se odkazem v Azure Data Factory a slouží transformaci dat. Při psaní této procedury je využit jazyk SQL, příkazem SELECT se získají data, provede se zamýšlená akce a výsledek operace se zapíše do cílové tabulky. V příkladě na obrázku 4.3 je akce spuštěna po překopírování dat z ADLS do Azure SQL Database a dopočítávají se hodnoty do sloupce velocityDelta pomocí Stored Procedure s názvem 'count_velocity_delta'. Tato operace využívá výpočetní výkon serveru, na kterém je provozována databáze, který je definován pomocí jednotek DTU, popsaných v kapitole 3.3.1.



Obrázek 4.2: Struktura ETL architektury, která je řízena pomocí Azure Data Factory.

Po těchto krocích je k dispozici datový základ. V další kapitole je přiblížen systém strojového učení Azure Machine Learning a vytvoření příkladu s využitím nasbíraných dat.



Obrázek 4.3: Schéma, které z nerelačního úložiště získá data do relačního modelu podle zadaného schématu a použije Stored Procedure 'count_velocity_delta' pro analytický výpočet.

Kapitola 5

Datová analýza a strojové učení

Předchozí kapitoly se věnují popisu variant datových úložišť a realizaci vybraného úložišťového řešení. V této kapitole jsou popsány komponenty služby Azure Machine Learning Services, které budou použiyt pro vytváření, správu a používání modelů strojového učení a umělé inteligence za použití nástrojů v jazyce Python. Ve druhé části kapitoly jsou popsány jiné metody, které se využívají při datové analýze. V další kapitole je ukázka využití této služby na ukázkovém příkladu.

5.1 Co je strojové učení?

Strojové učení (Machine Learning) je technika, která umožňuje výpočetním systémům využít existující data k popsání budoucího chování, trendů a výstupů sledovaného systému [25]. Použití strojového učení umožňuje vytvoření rozhodovacího systému bez nutnosti explicitně programovat každý prvek rozhodovacího algoritmu.

5.2 Machine Learning na platformě Azure

Azure Machine Learning (AML) je řešení integrované do platformy Azure, které umožňuje přípravu dat z datových úložišť, vývoj experimentů a vytvoření modelů a jejich využívání na cloudové službě [25].

Hlavní součásti AML jsou

- AML Workbench,
- AML Experimentation,

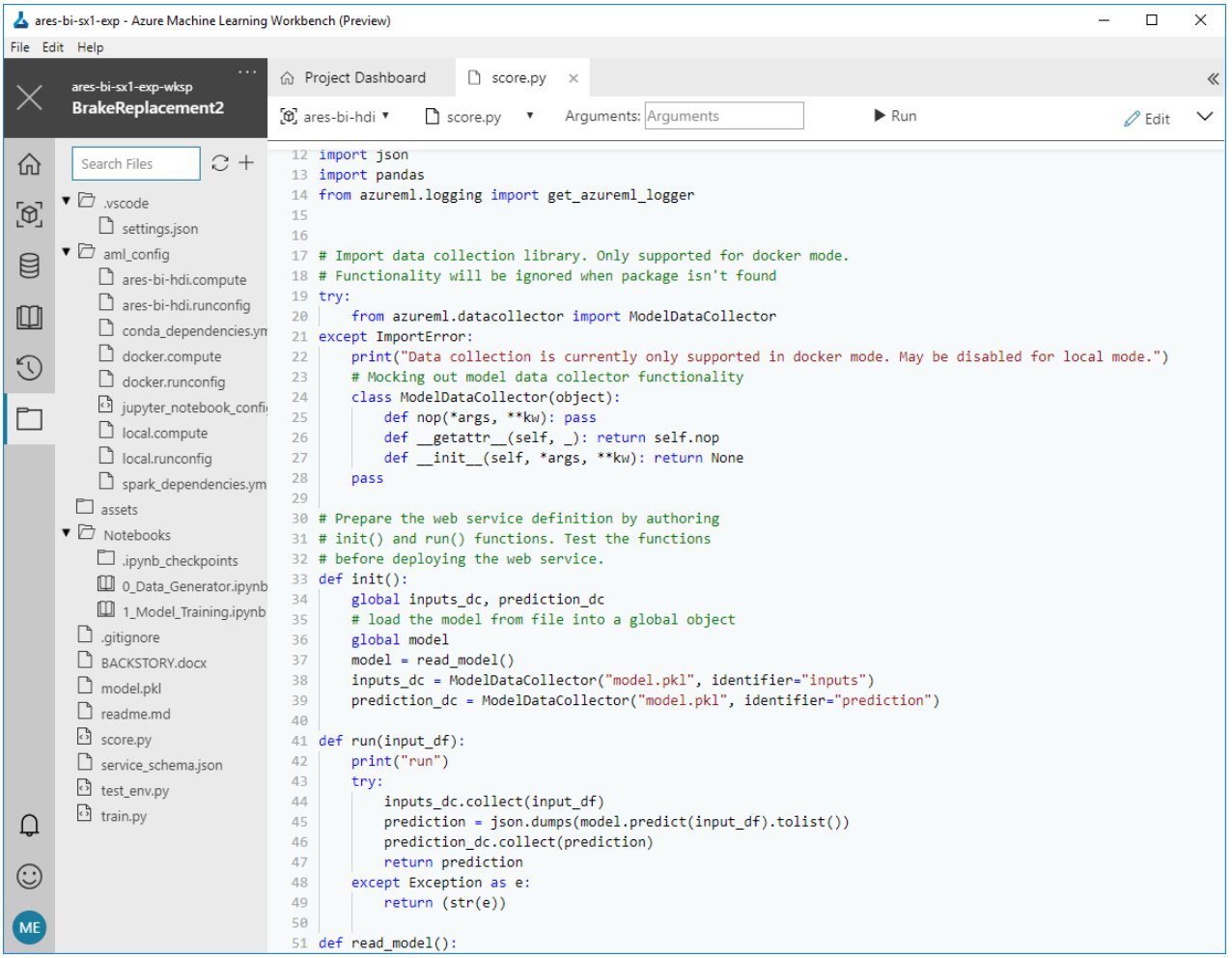
- AML Model Management,
- AML knihovny pro Apache Spark (MMLSpark Library),
- Visual Studio Code Tools.

Machine Learning na platformě Azure umožňuje využití open-source balíčků s frameworky strojového učení, které jsou často psány v jazyce Python, například scikit-learn, TensorFlow nebo Microsoft Cognitive Toolkit. Experimenty je možné provádět v různých výpočetních prostředích, například na platformě HDInsights, která je optimalizovaná pro paralelní výpočty datové analýzy.

5.3 Popis jednotlivých nástrojů pro vytváření a správu datových modelů

5.3.1 Azure Machine Learning Workbench

Azure Machine Learning Workbench je desktopová aplikace určená pro vývoj a běh experimentů strojového učení. Vytváří a obsahuje Machine Learning projekty, ve kterých jsou soubory pro vizualizaci, experimentování, přípravu datových zdrojů, předzpracování dat, vývoj skórovacího skriptu a delegace experimentů do různých výpočetních prostředí. Ukázka aplikace je na obrázku 5.1.



Obrázek 5.1: Ukázka studia Machine Learning Workbench s otevřeným skórovacím xskriptem, který je použit při vytváření modelu pro reálný příklad využití popsaný v kapitole 6.1.

5.3.2 Azure Machine Learning Experimentation Service

Tato služba se stará o provádění experimentů strojového učení. Umožňuje vytvoření a správu projektů ML, integraci s Git repositářem a kontrolu přístupů. Experimenty je možné spustit lokálně nebo je nahrávat do Spark Clusteru vytvořeného v prostředí HDInsights. Běhy experimentů jsou oddělené a reprodukovatelné, takže lze z historie vybrat model s nejlepšími výsledky.

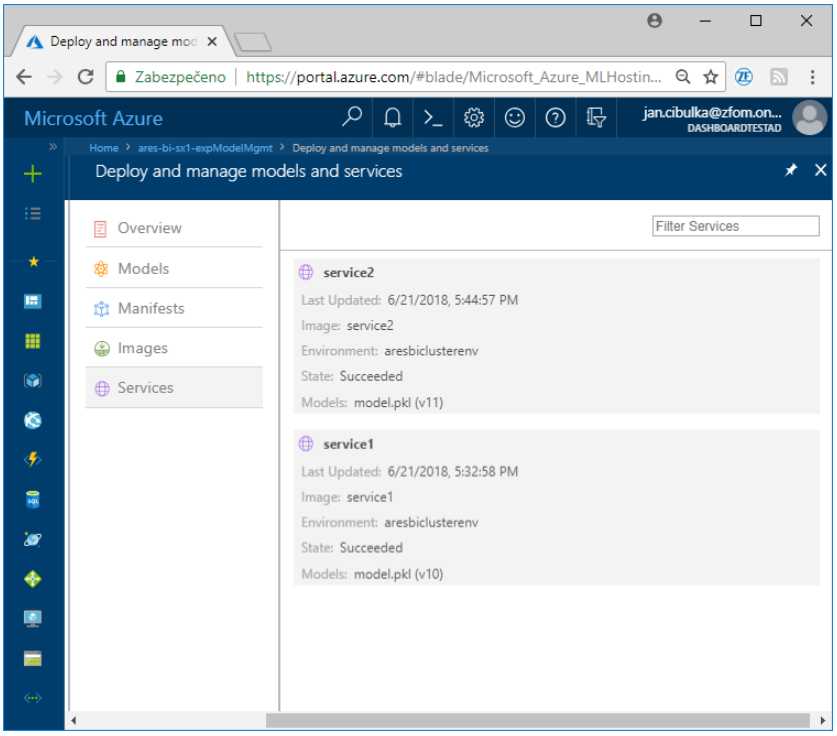
5.3.3 Azure Machine Learning Model Management Service

Do této služby se vybrané datové modely registrují a aplikují v různých prostředích a službách. Pro vytvoření webové služby s datovým modelem je potřeba několik souborů.

- Obraz prediktivního modelu. Tento obraz je výstupem trénovacího experimentu.
- Skórovací skript. Tento skript specifikuje, co se děje při využití služby. Obsahuje metody init() a run(), které definují akce při spuštění služby, nahrávání dat do modelu ze vstupu do webové služby a vytvoření návratové hodnoty modelu.
- Soubor se závislostmi, který obsahuje informace o odkazech na externí balíčky, využité v modelu nebo ve skórovacím skriptu.

Tyto tři soubory se zkombinují do aplikačního balíčku ('manifestu'), který obsahuje všechna data pro běh aplikace. Z manifestu se vytvoří obraz aplikace, který již je optimalizován z hlediska cílové platformy, na kterém poběží webová služba.

Obrazy aplikace jsou vytvořené pomocí platformy Docker. Ta je určena pro vývoj, vydávání a běh aplikací. Hlavní výhodou této platformy je zabalení obsahu do tzv. 'Docker Containeru', který pro jakýkoliv obsah a formu aplikace poskytuje rozhraní umožňující správu a běh aplikace v kompatibilním prostředí. Ukázka portálového rozhraní je uvedena na obrázku 5.2.



Obrázek 5.2: Ukázka Model Management na Azure portálu se dvěma běžícími službami.

5.3.4 Využívání webové služby

Po úspěšném zprovoznění webové služby za pomocí Azure Model Management na vybraném prostředí je možné do této služby posílat zprávy které obsahují vstupní data do modelu, na které služba odpoví zprávou s datovým výstupem z modelu.

Požadavek je většinou formulován externí aplikací ve formě HTTP POST zprávy na URL služby. V hlavičce POST zprávy musí být autorizační klíč k API jakožto bezpečnostní prvek. Vstup je pak uložen v samotném obsahu zprávy. Je možné mít více instancí vstupu a výstupu v jednom volání služby. Příklad testování služby z konzole v jazyce Bash je ukázán ve výpisech 5.1 a 5.2.

```
jan_cibulka@Azure:~$ curl -X POST
    -H  "Content-Type:application/json"
    -H "Authorization:Bearer SClFblwJSly6k90BdiI3uKDSLJmVDkHy"
    --data "{\"input_df\": [[2, 1427, 41, 2, 142778]]}"
     http://52.136.243.218/api/v1/service/service2/score
```

Výpis 5.1: Testovací příkaz v Bash konzoli. Curl je příkaz pro tvorbu webových HTTP požadavků. -X POST vyjadřuje, že se jedná o vkládání dat do služby. -H označuje prvky hlavičky zprávy, ve které je specifikván autorizační klíč a struktura obsahu. –data obsahuje řetězec se vstupními daty. Na poslední řádce požadavku je URL webové služby, která označuje webovou službu a konktrétní funkci, která bude vykonána.

```
"[1]"
```

Výpis 5.2: Odpověd ze služby ve formě řetězce. Pokud by byl do služby poslán vektor požadavků, toto pole by mělo prvek pro výsledek každé predikce.

5.4 Datová Analýza

Prostředí MS Azure nabízí několik nástrojů pro analýzu uložených dat. V následujících částech jsou popsány a porovnány dvě nejčastější služby využívané pro tento účel, Azure Analysis Services a Azure Data Lake Analysis.

5.4.1 Azure Analysis Services

Azure Analysis Services (AAS) je služba MS Azure ve formě analytické platformy. Umožňuje nahrávání dat z mnoha zdrojů, modelování vztahů mezi daty a počítání metrik. Metriky jsou agregační výpočty psané v jazyce DAX (Data Analysis Expressions)

[12], který obsahuje kolekci operátorů a funkcí. Ty se dají používat ve vzorcích k výpočtům a vrácení jedné nebo více hodnot. DAX je navržený pro práci s daty uloženými v tabulce a připomíná výrazy vytvářené v MS Excel.

Vývoj datového modelu probíhá ve Visual Studiu, ve kterém se vytvoří tabulární projekt. V tom se nakonfigurují datové zdroje a rozsah dat, který je udržován v datovém modelu. Ve Visual Studiu se pak vytvoří vztahy mezi daty, nové datové typy, nové metriky dat a uživatelské role omezující přístup k různým částem modelu. Dokončený tabulární model se nahraje do služby Azure Analysis Services.

Data jsou ve službě AAS uložena ve sloupcovém formátu, který je popsán u služby Azure Data Warehouse v kapitole 3.3.2 a je optimalizován pro rychlé dotazování a analýzu dat. Nevýhodou AAS je vysoká cena, která se odvíjí od vysoké rychlosti dotazování. Z tohoto důvodu je důležité zvážit, jak velký rozsah dat bude do služby nahrán a který bude analyzován. Většinou se používá model, ve kterém jsou všechna data trvale uložena v Azure Data Lake Store nebo Azure SQL Database a pouze nejaktuálnější data, která jsou předmětem analýz, se nahrají do AAS. Tato služba se hodí pro získání přehledu nad aktuálními daty. Pro širší analýzu historických dat se hodí spíše služba Azure Data Lake Analytics.

5.4.2 Azure Data Lake Analytics

Azure Data Lake Analytics (ADLA) [20] je služba pro vykonávání analytických operací, které zpracovávají Big Data. Využívá úložiště Azure Data Lake Store jako zdroj dat. Výsledky mohou být ukládány zpět do ADLS uložiště nebo převedeny do relačního modelu a vloženy do Azure SQL Database nebo Azure SQL Data Warehouse. ADLA přináší výhody cloud-computingu, kdy není třeba se starat o nastavení výpočetního serveru nebo clusterového prostředí, pouze se nastaví dotaz na data a požadovaný výpočetní výkon. Dotazy (queries) se vyvíjejí ve Visual Studio, které poskytuje nástroje pro nahrávání, debug a monitorování běhu dotazů v ADLA službě. Visual Studio vždy vygeneruje graf úlohy, kde je ukázán čas a využitý výkon jednotlivých fází procesu, což vývojářům pomáhá při optimalizaci dotazů.

ADLA využívá strukturovaný dotazovací jazyk U-SQL, který je rozšířením jazyka SQL o možnost využít vlastní funkce psané v jazycích C#, Python nebo R a další rozšíření programovatelnosti dotazů.

Při nahrání úlohy se specifikují tři věcí - U-SQL skript, zdroj dat pro skript a počet Analytics Unit (AU) rezervovaných pro exekuci skriptu. U-SQL kompilátor a optimalizátor vyhodnotí skript a data a vytvoří plán, jak vyřešit úlohu. Tento plán je rozdělen

do malých úkolů, kterým se říká 'vertex'. Nejjednodušší úlohy obsahují pouze jeden vertex, složité úlohy jich mohou mít tisíce. Zároveň jsou založeny výpočetní jednotky AU, které mají v současnosti dvoujádrový procesor a 6 GB paměti RAM. Určí se pořadí, ve kterém je nutné vertexy vykonávat a ty se dále rozdělí mezi AU, které je vyřeší. Pokud je zvoleno více AU, je práce optimalizována tak, aby běželo maximum úloh paralelně jak jen to možné. Cena za využití služby se vypočítá na základě alokace AU každou sekundu, s cenovým základem 2$ za hodinu pro 1 AU.

5.4.3 Porovnání služeb Azure Analysis Services a Azure Data Lake Analytics

Obě tyto služby jsou využity pro zpracování a analýzu dat. Azure Data Lake Analytics se zaměřuje na analýzu dat uložených v nestrukturovaném formátu v Azure Data Lake Store úložišti. Protože je Azure Data Lake Store velmi levnou variantou úložiště, hodí se Data Lake Analytics pro zpracovávání velkého objemu dat. Proto je ideální pro počítání statistik historických dat. Výstupem této služby jsou uložená statistická data do relačního nebo nerelačního úložiště. Oproti tomu Azure Analysis Services není vhodný pro analýzu historických dat, více se orientuje na interaktivní modelování aktuálních dat a přípravě dat pro vizualizační nástroje.

Kapitola 6

Vytvoření prediktivního modelu pro reálný příklad

V této kapitole je pro účely vytvoření příkladu s využitím popsaných služeb a nasbíraných a uložených dat vytvořen klasifikační model za použití nástrojů popsaných v kapitole 5.2. Data a výsledky jsou pak vizualizovány v jednoduchém reportu popsaného v kapitole 6.8.

6.1 Návrh modelu

6.1.1 Motivace a popis modelu

Příklad je zaměřen na optimalizaci plánování návštěvy servisu za účelem výměny brzdových destiček. Využita jsou jízdní data, data o návštěvách servisů a statická data o servisech a vozidlech. Model se ze stylu jízdy a ujetých kilometrů pokusí odhadnout, v jakém stavu jsou brzdové destičky, zda potřebují vyměnit nebo za jak dlouho bude výměna potřeba.

6.1.2 Vstupní data modelu

Vstupní data do modelu obsahují

- Typ nainstalované brzdy ('brakeType'),
- ujeté kilometry od poslední návštěvy servisu, při které došlo k výměně brzd ('mileage'),

- počet velkých změn rychlosti (prudké brzdění) vypočítané z jízdních dat,
  - o 30-40 km/h ('dccel1'),
  - o 40-50 km/h ('dccel2'),
  - o více než 50 km/h ('dccel3').

Jízdní data s údajem o rychlosti mají periodu vzorkování tři sekundy. Uvedená velikost se týká změny rychlosti ve dvou sousledných zprávách. Cílem je získat vstupní vektor ve formě $[brakeType; mileage; dccel1; dccel2; dccel3]$.

6.1.3 Výstup modelu

Model má rozhodnout o tom, **zda jsou v současném stavu brzdové destičky v pořádku, nebo je třeba navštívit servis**. Výstup je tedy ve formě ano/ne. Výstupní veličina je pracovně nazvána 'isOk', hodnota '1' značí, že jsou brzdy v pořádku, hodnota '0' značí, že je potřeba navštívit servis.

6.1.4 Příprava dat

Množina dat, využitá při trénování klasifikátoru, má navíc sloupec s požadovaným výstupem (isOk), aby mohla být využita technika trénování s učitelem, které je přiblížena v 6.2. Tato množina dat je připravena následujícím způsobem.

1. Data jsou rozdělena podle vozidel a na úseky rozdělené podle času návštěv servisů s výměnou brzdových destiček. Všechny události, kdy došlo k výměně brzdových destiček, představují jeden trénovací vstup s hodnotou 'isOk = 0'. Pokud došlo k návštěvě servisu, ale nebyla provedena výměna brzdových destiček, také se jedná o trénovací vstup, který ovšem má 'isOk = 1'. Také se ze servisních zpráv zjistí typy brzd, s jakými auto dorazilo a odjelo.
2. Určení počtu ujetých kilometrů pro daný časový úsek.
3. Výpočet velikosti brzdění z jízdních záznamů pro daný časový úsek.
4. Identifikace brzdění, které odpovídají uvedeným třem skupinám. Uložení těchto událostí do tabulky Events, popsané v 4.5.
5. Výpočet četnosti brzdění pro všechny tři typy (dccel1, dccel2 a dccel3) pro daný časový úsek a vozidlo z tabulky Events, uložené do tabulky s vozidly, Ref_Vehicle.

Tabulka 6.1: Ukázka dat, připravených jako vstup pro algoritmus strojového učení.

Id	deviceId	brakeType	dc1	dc2	dc3	mileage	isOk
1	7C9763002114	3	1345	37	4	177261	1
2	7C9763002246	2	412	24	4	149831	1
3	7C9763002115	1	671	50	7	138965	0
4	7C9763002227	3	460	29	6	102016	1
5	7C9763002112	3	681	52	6	123989	1
6	7C9763001FF9	1	977	21	6	113706	1

6. Z informací z tabulky Ref_Vehicle se k jednotlivým vozidlům vygenerují vstupní vektory pro klasifikační model do tabulky ML_Brake_Service.

Vzorek množiny použité pro trénování klasifikátoru je v tabulce 6.1.

6.2 Výběr typu modelu

Po definování formátu vstupních a výstupních dat v části 6.1 je třeba vybrat model, který bude vhodně reprezentovat rozhodovací algoritmus. V této práci je použit balíček scikit-learn [19], [7], který poskytuje nástroje pro datovou analýzu, strojové učení, dolování informací z dat ('data mining'), dokumentaci k jednotlivým algoritmům učení a je pod otevřenou licencí. Další informace o strojovém učení jsou získány z průvodce výběru algoritmu pro strojové učení z dokumentace k Microsoft Azure. [10]. V této části je uvedeno shrnutí základních přístupů při vytváření modelů strojového učení.

Ve strojovém učení je k dispozici trénovací množina a vybraný učící algoritmus se snaží nastavit model tak, aby reagoval na trénovací množinu vybraným způsobem. Základní dělení je na strojové učení s učitelem ('supervized') a bez učitele ('unsupervized'). Ve strojovém učení bez učitele nemají trénovací data informace o cílové hodnotě nebo zařazení. Cílem učení bez učitele je získat náhled na uspořádání trénovací množiny dat a nebo popsat její strukturu. Základním přístupem ve strojovém učení bez učitele je shlukování objektů do skupin, například pomocí algoritmu k-means. Ve strojovém učení s učitelem jsou trénovací data tvořeny párem vstupního vektoru a požadovaného výstupu. Algoritmus se snaží předvídat výstupní hodnotu na základě vstupních záznamů. Při správném natrénování modelu je chyba předvídání hodnot na trénovací množině minimální a model je připraven pro hodnocení reálných dat. Rozlišují se dvě základní skupiny algoritmů určené pro učení s učitelem: regrese a klasifikace.

6.2.1 Regrese

Algoritmy s regresí jsou založeny na určování neznámé hodnoty atributu objektu popsaného vstupními daty. Hodnota tohoto atributu je spojitá. Existuje mnoho druhů algoritmů s regresí, základní myšlenkou vytvořit kombinaci vstupních dat takovou, aby výsledkem byla požadovaná výstupní hodnota.

Základním přístupem využívaným v regresi je lineární regrese. V této skupině jsou algoritmy, ve kterých je cílená hodnota lineární kombinací vstupních hodnot. Mezi hlavní algoritmy v této skupině patří metoda nejmenších čtverců a bayesovský přístup v metodě maximální věrohodnosti, obě s mnoha variantami.

Mezi další druhy algoritmů pro regresi patří Support Vector Regression, Regrese pomocí nejbližšího souseda, regresní rozhodovací stromy a regresní vícevrstvé perceptronové sítě.

6.2.2 Klasifikace

V klasifikaci jsou data použita pro identifikaci kategorie, do které objekt patří. Rozdíl oproti regresi je ten, že výstupní hodnota není spojitá, ale identifikuje kategorii. Rozlišují se úlohy pro klasifikaci mezi dvěma nebo více třídami. Dále jsou uvedeny některé základní metody v klasifikačních úlohách.

Diskriminační analýza

Tato metoda se používá pro rozdělení vstupních objektů mezi konečný počet tříd. Hlavními variantami této metody jsou lineární a kvadratická diskriminační analýza. V nich jsou třídy charakterizovány Gaussovým rozdělením v dimenzích vstupního vektoru. Z trénovací množiny jsou určeny tvary těchto Gaussových rozložení pomocí určení střední hodnoty, variance a relativní četností dané třídy. Vstupní vektory jsou pak přiřazeny do třídy po vyhodnocení Bayesova pravidla.

Logistická regrese

Logistická regrese je i přes své jméno lineárním modelem určeným pro klasifikaci mezi dvě třídy. Více o Logistické regresi v kapitole 6.3.

Klasifikační rozhodovací stromy

Cílem klasifikačních rozhodovacích stromů je vytvořit model, který určuje výstupní hodnotu pomocí vyhodnocování posloupnosti pravidel ze vstupních dat. Rozhodovací stromy bývají jednoduše vizualizovatelné a interpretovatelné. Umožňují klasifikaci mezi konečný počet tříd. Jejich nevýhodou bývá časté

přetrénování, kdy má strom příliš mnoho pravidel, které vyhovují trénovací množině dat, ale špatně popisují obecná data.

Perceptronová síť

Vícevrstevná perceptronová síť je algoritmus učení s učitelem, ve kterém se algoritmus učí nelineární funkci $f(.) : R^i \rightarrow R^o$, kdy R^i a R^o označují vstupní a výstupní vektory. Perceptrony jsou rozděleny do vrstev, používá se vstupní a výstupní vrstva, mezi kterými se nachází skryté vrstvy. Každý perceptron ve skrytých vrstvách transformuje hodnoty z výstupu perceptronů předchozí vrstvy pomocí vážené lineární sumy. Tato hodnota je transformována nelineární aktivační funkcí $g(.)R \rightarrow R$, jejíž výsledek je na výstupu z neuronu. Výhodou perceptronových sítí je možnost vytvářet velmi flexibilní lineární i nelineární modely. Nevýhodami perceptronové sítě je velké množství parametrů při ladění sítě, iterativní přístup trénovaní může nalézt suboptimální řešení a algoritmus je citlivý na škálování vstupních dat.

6.2.3 Vlastnosti algoritmů strojového učení

Jednotlivé algoritmy jsou hodnoceny podle jejich kvalitativních vlastností, které jsou v této části přiblíženy.

Přesnost

Přesnost je samozřejmě hlavním ukazatelem úspěšnosti modelu, ne vždy je však vhodné dosažení nejvyšší možné hodnoty. Někdy postačí aproximace, která má kratší trénovací dobu. Aproximace také bývají obecnější a jsou odolnější vůči přetrénování.

Trénovací doba

Trénovací doba je velmi odlišná pro jednotlivé algoritmy, například iterativní trénování perceptronové sítě zpravidla trvá déle než použití metody nejmenších čtverců na stejné velkosti trénovací množiny.

Linearita

Lineární algoritmy předpokládají, že jsou data oddělená přímkou (nebo ekvivalentem vyšší dimenze). Příkladem je lineární regrese nebo logistická regrese. Takové algoritmy mohou v určitých případech zkrátit dobu trénování, ale v nevhodném případě, kdy je vhodný nelineární popis dat, velmi sníží výslednou přesnost modelu.

Počet parametrů algoritmu

Každý algoritmus je konfigurován pomocí sady parametrů. Při ladění algoritmu se často experimentuje s různými nastaveními algoritmu a zkoumá se, jakým způsobem ovlivňují přesnost a trénovací dobu. Menší počet parametrů

znamená, že je potřeba vyzkoušet méně variant algoritmu, vyšší počet zase umožňuje nastavení algoritmu na míru daného problému.

Dimenze vstupních dat

U některých druhů dat je počet prvků ve vstupních vektorech velký v porovnání s počtem vstupů. Vysoký počet vstupů může velmi prodloužit dobu trénování některých algoritmů. Řešením může být redukce dimenze vstupního vektoru pomocí nástrojů z balíčku scikit-learn nebo použití algoritmu Support Vector Machines, je navržen pro práci s vyšší dimenzí vektoru vstupních dat.

Po uvedení několika základních přístupů ve strojovém učení je třeba vybrat učící algoritmus. Kvůli povaze navrženého systému je vybrána klasifikace do dvou tříd v podobě logistické regrese. Algoritmus logistické regrese je přiblížen v následující kapitole.

6.3 Logistická regrese

6.3.1 Popis algoritmu

Logistická regrese je algoritmus strojového učení, který původně vychází z matematické statistiky. Je to standardní metoda pro řešení problému klasifikace do dvou tříd. Pro klasifikaci se používá logistická (sigmoidální) funkce s předpisem

$$f(x) = \frac{1}{1 + e^{-x}}, \tag{6.1}$$

kde e je Eulerovo číslo a x je funkční proměnná. Výsledná křivka připomíná písmeno 'S'.

Logistická regrese, podobně jako lineární regrese, využívá pro svoji reprezentaci rovnici. Vstupní data X jsou lineárně kombinována za pomocí koeficientů β a výstupem je hodnota y. Na rozdíl od lineární regrese je však y z množiny $\{0, 1\}$ a není to spojitá hodnota. Reprezentace modelu, který je uložen a obsažen ve webové službě, obsahuje koeficienty β.

Model logistické regrese určuje pravděpodobnost třídy 1, respektive pravděpodobnost, že vstup X má za výsledek $y = 1$ a nebo $y = 0$, jak je ukázáno v následujících rovnicích.

$$P(y = 1|X) = \frac{1}{1 + e^{\beta^T x}} = 1 - P(y = 0|X), \tag{6.2}$$

$$P(y = 0|X) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}} = 1 - P(y = 1|X). \tag{6.3}$$

Podíl rovnic (6.2) a (6.3) je zajímavý, protože odpovídá podílu šance, že výstup je $y = 1$ a šance, že výstup je $y = 0$.

$$\frac{P(y=1|X)}{P(y=0|X)} = \frac{\frac{1}{1+e^{\beta^T x}}}{\frac{e^{\beta^T x}}{1+e^{\beta^T x}}} = e^{\beta^T x} \tag{6.4}$$

Uvedený podíl má obor hodnot $H =< 0, \infty >$. Pokud jsou logaritmovány obě strany rovnice (6.4), je získána rovnice (6.5), která má na levé straně logaritmus šance původní funkce v oboru hodnot $H =< -\infty, \infty >$ a na pravé straně je lineární transformace vstupního vektoru X.

$$ln\left(\frac{P(y=1|X)}{P(y=0|X)}\right) = \beta^T X \tag{6.5}$$

Z uvedeného vztahu je již zřejmé, že se jedná o lineární binární klasifikátor, který může být zapsán jako

$$\begin{aligned} y = 0 &: \beta^T X < 0, \\ y = 1 &: \beta^T X > 0. \end{aligned} \tag{6.6}$$

6.3.2 Strojové učení s logistickou regresí

Balíček scikit-learn poskytuje iterativní optimalizační metody (solvery) k nalezení optimálního vektoru β s názvy liblinear,newton-cg, sag a lbfgs. Všechny tyto metody jsou různými variantami gradientního postupu při řešení optimalizačního problému.

$$\min_{\beta} \left\{ \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n} ln(1 + e^{-y_i(\beta^T X)}) \right\}. \tag{6.7}$$

Výraz (6.7) obsahuje minimalizaci součtu dvou členů podle parametru β. První člen sleduje velikost vektoru parametrů β. Druhý člen v sobě akumuluje chybu klasifikace trénovací množiny. Správně klasifikované hodnoty přispívají do sumy malou mírou, hodnoty nejistě nebo špatně klasifikované přispívají do sumy více. X je vstupní vektor a y je výstup s hodnotami z množiny $\{0, 1\}$. Hodnota C je učící parametr a udává poměr důležitosti mezi velikostí vektoru β a druhým členem při minimalizaci kritéria.

Při trénování modelu logistické regrese jsou algoritmu poskytnuta trénovací data a parametry nastavení modelu, mezi které patří C a druh solveru.

6.4 Popis vybraného modelu

Při experimentování s modelem byl vybrán jako parametr solver 'newton-cg' a hodnota C byla zvolena 10.0. Výsledná přesnost modelu je 93.45%. Vizualizace klasifikací reálných dat je ukázána v části 6.8.6.

6.5 Obsluha modelu

Vybraná varianta modelu byla uložena a pomocí postupu popsaného v kapitole 5.3.3 o Model Management byla vytvořena webová služba obsahující tento model.

Pomocí Azure Data Factory je vytvořen mechanismus, který při každé aktualizaci vozidla vloží záznam s novými daty do tabulky Ml_Brake_Service. Zároveň je vytvořen jednoduchý skript v prostředí Azure Function, který přečte všechny nevyřízené záznamy z této tabulky, zavolá webovou službu s modelem, nechá nová data klasifikovat a záznam zapíše zpět do tabulky Ml_Brake_Service. V tabulárním modelu v SQL databázi pak existují uložené procedury, které s novými výsledky klasifikací přepočítají existující statistiky vozidel.

6.6 Popis vytváření predikcí

Vytvořený model umožňuje klasifikaci současného stavu. Je však zájem o predikci možného budoucího stavu, který je odhadnut za použití modelu následujícím způsobem. Ze současných hodnot se vypočítá předpokládaná četnost sledovaných prudkých brzdění za δ km pomocí lineárních rovnic,

$$n_\delta = \frac{n}{m}\delta + n \tag{6.8}$$

kde n_δ je počet událostí daného typu za δ km, n je současný počet událostí, m_δ je počet ujetých km za δ km a m je současný počet ujetých kilometrů. Vypočítané hodnoty se pak přidají do tabulky Ml_Brake_Service, kde je zaznamenána velikost δ, aby nedošlo k záměně s reálnou množinou návštěv servisů.

6.7 Shrnutí činnosti Azure Data Factory

Protože je celý systém řízen pomocí instance Azure Data Factory, je vhodné si shrnout akce, které jsou vykonávány při sběru dat, přípravě dat, vyhodnocování požadavků a prezentací. Aplikace se pouští periodicky a interval je možné nastavit dle potřeby. Při spuštění aplikace se provedou následující akce:

1. Jsou nahrána nová servisní a jízdní data z nestrukturovaného úložiště do SQL databáze. Jednotlivé textové zprávy jsou upraveny tak, aby vyhovovaly datovým typům relačního modelu.
2. Je vypočítána veličina VelocityDelta, která udává změnu rychlosti mezi záznamy. Pro tuto činnost je využita platforma HDInsights se strukturou optimalizovanou pro výpočetní operace.
3. Jsou identifikovány jízdní události. Hodnoty VelocityDelta větší než 30 jsou roztříděné do tří pásem a uloženy do tabulky Events.
4. Ze získaných dat jsou přepočítány statistiky vozidel v tabulce Ref_Vehicle.
5. Pro každé vozidlo jsou vytvořeny vstupní vektory pro model strojového učení. Používá se současný stav a predikce stavu za 10 až 100 tisíc kilometrů. Data jsou uložena v tabulce Ml_Brake_Service, která je periodicky kontrolována a nové vstupy jsou odesílány do klasifikátoru a ukládány zpět do této tabulky jako vyřízené požadavky. Při trénování modelu jsou vypuštěny záznamy získané pomocí predikce.
6. Když jsou požadavky všech vozidel vyřízeny, přepíší se výsledky klasifikace v databázi novými záznamy a provede se obnova vizualizací.

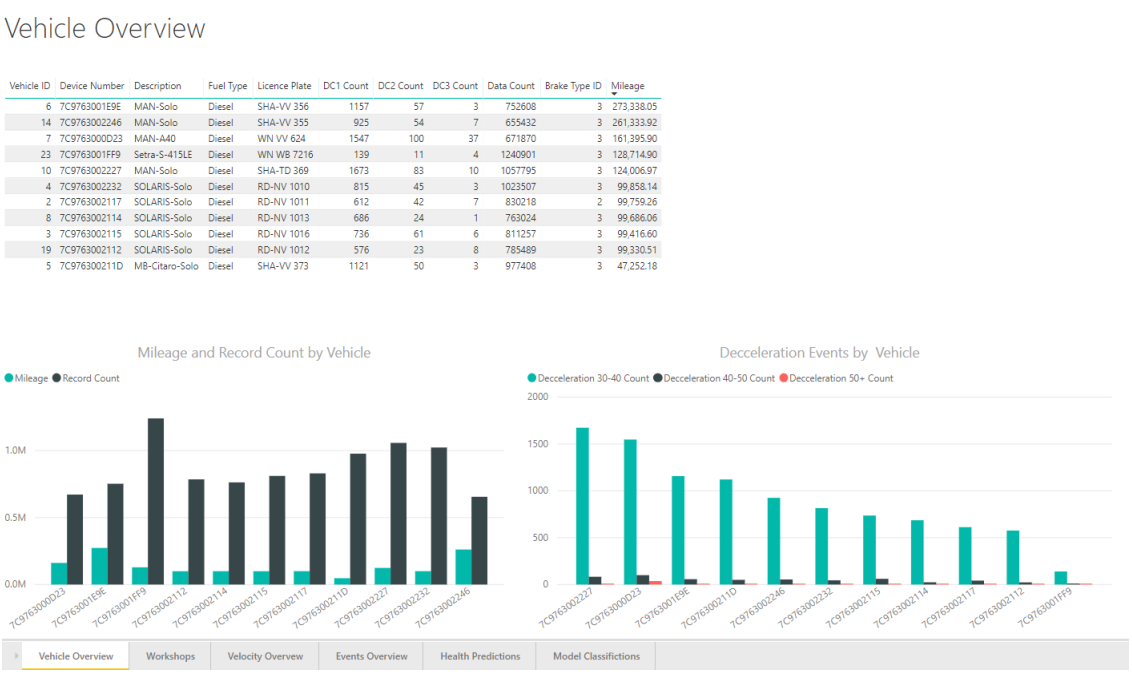
6.8 Vizualizace dat

V této části je ukázán report s vizualizacemi rozdělenými do několika částí, které obsahují data, se kterými se pracovalo v této práci. Reporty jsou vytvářeny pomocí služby PowerBI, která obsahuje nástroje pro tvoření vizualizací a základní datovou analýzu. Reporty jsou pak k dispozici na webovém portálu, v mobilním zařízením nebo je možné je exportovat a vložit na vlastní webovou stránku. Zároveň je přístup k reportům snadno regulován pomocí definování uživatelských skupin, které mají přístup k různým prvkům reportů. Reporty jsou tvořeny ve studiu Power BI Desktop.

Je vytvořeno šest skupin vizualizací popsaných a ukázaných v následujících částech: Přehled vozidel, Přehled servisů, Přehled rychlosti, Přehled událostí, Predikce zdraví a Přehled klasifikace modelu.

6.8.1 Přehled vozidel

Nejdůležitější skupinou vizualizací je přehled vozidel a jejich aktivity, který je ukázán na obrázku 6.1. Je zde uvedeno, kolik které vozidlo ujelo celkem kilometrů, jaký je celkový počet přijatých zpráv a kolik jízdních událostí spojených s brzděním bylo identifikováno.



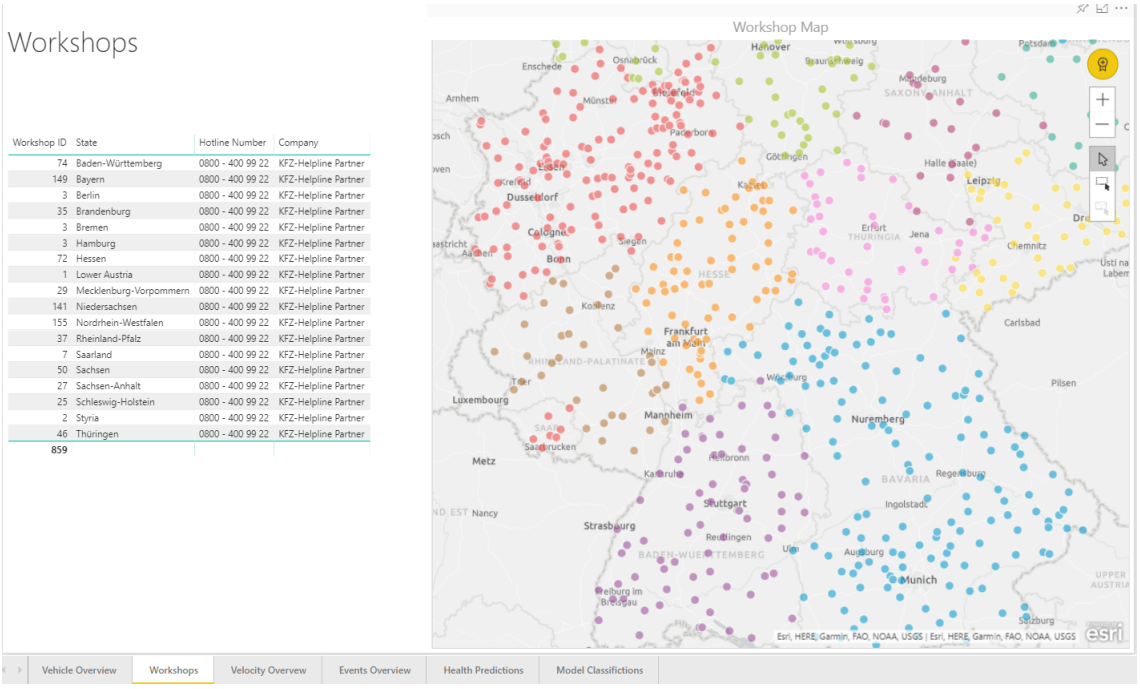
Obrázek 6.1: Vizualizace přehledu vozidel, který obsahuje seznam aktivních vozidel a jejich základní statistiky.

6.8.2 Přehled servisů

Tento zahrnuje základní informace o servisech. Na obrázku 6.2 je ukázána mapa, která zobrazuje spolupracující servisy v Německu s barevným rozlišením podle spolkových zemí.

6.8.3 Přehled rychlosti

Další část reportu obsahuje vizualizační prvky zobrazující jízdní rychlosti a zrychlení/zpomalení při jízdě.



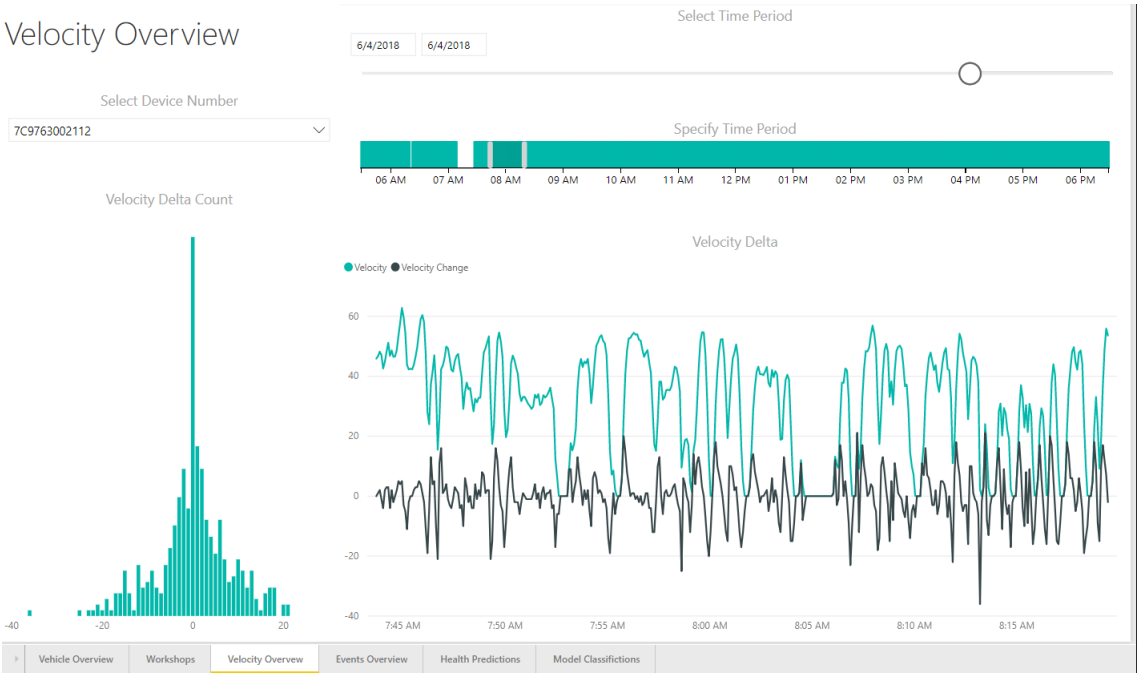
Obrázek 6.2: Vizualizace obsahující mapu s rozmístěním servisů.

V obrázku 6.3 je ukázáno několik věcí:

1. Komponenta pro výběr časového intervalu, který omezuje rozsah dat v ostatních grafech.
2. Tabulka s přehledem vozidel, ve které je možné zúžit výběr na jedno vozidlo a prozkoumat grafy pouze s jeho daty. Zároveň je zde uveden počet jízdních záznamů pro vozidlo a největší zpomalení vozidla.
3. V horním grafu je průběh rychlosti ve zvoleném časovém intervalu pro zvolené vozidlo.
4. V dolním pravém grafu je průběh změny rychlosti ve zvoleném časovém intervalu pro zvolené vozidlo.
5. V dolním levém grafu je histogram zpomalení a zrychlení ve zvoleném časovém intervalu pro zvolené vozidlo. Za velikost jednoho úseku horizontálního členění grafu je zvolena změna rychlosti mezi záznamy o 1 km/h.

6.8.4 Přehled událostí

Vizualizace, ukázané na obrázku 6.4, jsou zaměřené na přehled definovaných událostí. V části 6.1 jsou definovány tři základní události při jízdě, které jsou analyzovány. Těmito událostmi jsou brzdění o určitou hodnotu tříděné do tří pásem.



Obrázek 6.3: Ukázka vizualizace obsahující přehled rychlosti.

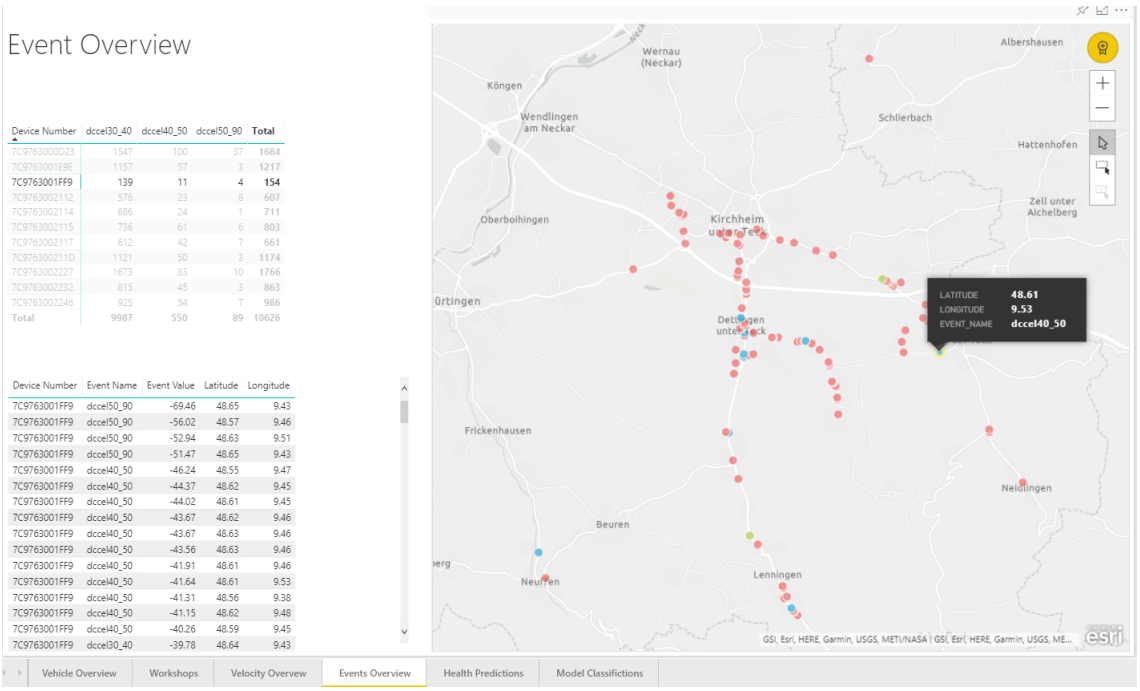
Události je možné filtrovat podle typu, času a vozidla. V dalších verzích a příkladech je možné volně přidávat nové druhy událostí pro další veličiny.

6.8.5 Přehled zdraví

V této vizualizaci je ukázáno, jaké je současné zdraví aktivních vozidel a odhad, jak se bude vyvíjet stav zdraví až do ujetí dalších 100 000 kilometrů.

6.8.6 Přehled klasifikace modelu

Ve vizualizaci, která je ukázaná na obrázku 6.6, jsou zobrazená data z tabulky Ml_Brake_Service, která popisují všechny případy využití modelu ke klasifikaci současného stavu nebo stavu vytvořeným pomocí predikce. Je možné vybrat všechna nebo jednotlivá vozidla. Ve 2D grafech je znázorněno hodnocení klasifikátoru pro každý typ události (dccce1,dcce2 a dccel3) a podle ujetých kilometrů. V grafu 'Average health by brake type' je zobrazeno průměrné zdraví brzd podle jejich typu a to včetně uvážení prediktivních případů.V tabulce 'All model classifications' je seznam všech záznamů se vstupními i výstupními daty.



Obrázek 6.4: Ukázka vizualizací obsahující mapu s přehledem jízdních událostí a tabulku s daty k jednotlivým událostem.

Obrázek 6.5: Ukázka vizualizací obsahující tabulku s odhadovaným zdravím jednotlivých vozidel a graf odhadu počtu zdravých vozidel v následujících kilometrech.

Obrázek 6.6: Ukázka vizualizací obsahující přehled hodnocení klasifikátoru.

Závěr

V této práci bylo popsáno využití platformy Microsoft Azure pro uložení, analýzu a vizualizace velkého množství dat. Motivace k vytvoření této diplomové práce vychází z příležitosti podílet se na nových projektech firmy OPENMATICS s.r.o, která se zaměřuje na oblast sběru telematických dat v automotive a logistice.

Na začátku práce probíhá seznámení s firmou OPENMATICS, s jejími produkty a s cloudovou platformou Microsoft Azure. Jsou popsány vlastnosti, které musí splňovat systém pro zpracování velkého objemu dat a komponenty, ze kterých je tvořen. Je navrhnut koncept systému, který přijímá servisní a jízdní data z vozidel, efektivně je ukládá, analyzuje je a prezentuje výsledky pomocí vizualizací.

V průběhu práce jsou popisované koncepty ilustrovány na reálném příkladu zpracování dat. Příklad je založen na sběru jízdních a servisních dat, jejich ukládání, analýze a vizualizaci. Analýza zde spočívá v jednoduché aplikaci strojového učení na základě reálných dat. Je natrénován klasifikátor, který z dat odhaduje stav brzdových destiček a generuje doporučení, zda je potřeba navštívit servis či nikoliv.

První část práce se věnuje popisu a porovnání způsobů, jakým ukládat data v úložištích na platformě MS Azure. Základní rozdělení úložišť je na úložiště strukturovaná a nestrukturovaná. Každý z těchto druhů se hodí pro ukládání jiného druhu vstupních zpráv a v možnostech, které nabízejí pro budoucí analýzu uložených dat. Pro zmíněný příklad je vytvořena kombinace strukturovaných i nestrukturovaných úložišť a jsou vytvořeny nástroje, které je obsluhují.

Ve druhé části práce je popsán systém Azure Machine Learning, který poskytuje nástroje pro přípravu, trénování, archivaci a využívání modelů strojového učení. V rámci příkladu je zde vytvořen klasifikační model, který predikuje stav brzdových destiček reálného vzorku vozidel na základě ujetých kilometrů, množství brzdění a statických údajů o vozidle. Tento model je pak využíván ve formě webové služby. Jízdní data, data ze servisů a výsledky analýz a klasifikace natrénovaného modelu jsou vizualizovány v reportu vytvořeným ve službě Power BI.

V blízké budoucnosti je plánováno širší využití vytvořeného systému pro analýzu servisních a jízdních dat. Zatím se podařilo nasbírat pouze část plánovaného rozsahu dat. Je třeba vymyslet další příklady využití servisních dat. Kvalita a provedení prediktivních modelů se mění s tím, jak kvalitní a rozmanitá jsou získaná data.

Literatura

[1] Acid, 2014. Blog uživatele 'aristote'. Dostupné z: https://blog.root.cz/aristote/acid/.

[2] Diagnostic trouble codes explained, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: https://www.obdautodoctor.com/scantool-garage/diagnostic-trouble-codes-explained.

[3] Structured query language/sql: The standard iso iec 9075 and various implementations, 2017. Dostupné z: https://en.wikibooks.org/wiki/Structured_Query_Language/SQL:_The_Standard_ISO_IEC_9075_and_various_Implementations.

[4] Azure database sql calculator, 2018. Webová aplikace společnosti Microsoft. Dostupné z: https://dtucalculator.azurewebsites.net/.

[5] Bosquez, L. Introduction to azure cosmos db: Graph api, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: https://docs.microsoft.com/cs-cz/azure/cosmos-db/graph-introduction.

[6] Boyko, A. Azure storage performance, 2015. Dostupné z: https://www.slideshare.net/AntonBoyko/azure-storage-performance,str.9.

[7] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122.

[8] Deng, S. What is azure sql data warehouse?, 2018. Dostupné z: https://docs.microsoft.com/cs-cz/azure/sql-data-warehouse/sql-data-warehouse-overview-what-is.

[9] Ellingwood, J. An introduction to big data concepts and terminology, 2016. Dostupné z: https://www.digitalocean.com/community/tutorials/an-introduction-to-big-data-concepts-and-terminology.

[10] Ericson, G. How to choose algorithms for microsoft azure machine learning, 2017. Dokumentace produktu Microsoft Machine Learning Studio. Dostupné z: https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice.

[11] Gara, R. Azure storage scalability and performance targets, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: https://docs.microsoft.com/cs-cz/azure/storage/common/storage-scalability-targets.

[12] Iseminger, D. Základy dax v power bi desktopu, 2018. Dokumentace produktu Microsoft Power BI Desktop. Dostupné z: https://docs.microsoft.com/cs-cz/power-bi/desktop-quickstart-learn-dax-basics.

[13] Lyon, R. Comparing azure data lake store and azure blob storage, 2018. Dokumentace produktu Microsoft Azure. Dostupné z: https://docs.microsoft.com/en-ca/azure/data-lake-store/data-lake-store-comparison-with-blob-storage.

[14] Mallon, A. What is a dtu?, 2017. Dostupné z: https://sqlperformance.com/2017/03/azure/what-the-heck-is-a-dtu.

[15] Mike Wasson, Rick Anderson, T. D. Data storage options, 2014. Dokumentace produktu Microsoft Azure. Dostupné z: https://docs.microsoft.com/en-us/aspnet/aspnet/overview/developing-apps-with-windows-azure/building-real-world-cloud-apps-with-windows-azure/data-storage-options.

[16] Mike Wasson, Z. T. Traditional relational database solutions, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/index.

[17] Ortloff, R. Azure sql data warehouse - massively parallel processing (mpp) architecture, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: https://docs.microsoft.com/en-us/azure/sql-data-warehouse/massively-parallel-processing-mpp-architecture.

[18] Painchaud, A. 4 steps to building an awesome big data solution on microsoft azure, 2018. Dostupné z: https://www.sherweb.com/blog/building-big-data-solution-azure/.

[19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and

Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[20] Reddy, S. Co je azure data lake analytics?, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/cs-cz/azure/data-lake-analytics/data-lake-analytics-overview`.

[21] Sarosh, R. Azure cosmos db hierarchical resource model and core concepts, 2018. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/en-us/azure/cosmos-db/sql-api-resources`.

[22] Shanan, R. Understanding block blobs, append blobs, and page blobs, 2018. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/en-us/rest/api/storageservices/understanding-block-blobs--append-blobs--and-page-blobs`.

[23] Vijayasarathy, S. What is azure event hubs?, 2018. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-about`.

[24] Wasson, M. Big data architectures, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/`.

[25] Winkler, M. What is machine learning?, 2017. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/en-us/azure/machine-learning/service/overview-what-is-azure-ml`.

[26] Wooley, T. Overview of azure data lake storage gen1, 2018. Dokumentace produktu Microsoft Azure. Dostupné z: `https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-overview`.