Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Bakalářská práce

Prediktivní analýza obsazenosti prostor s pomocí dat z WiFi přístupových bodů

Apti Archakov

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Apti Archakov

Systémové inženýrství a informatika
Informatika

Název práce

Prediktivní analýza obsazenosti prostor s pomocí dat z WiFi přístupových bodů

Název anglicky

Predictive analysis of space occupancy using data from WiFi access points

---

Cíle práce

Hlavním cílem práce je pomoci dat z univerzitních wifi přistupových bodů, predikovat obsazenost prostor v areálu ČZU.

Dílčí cíle práce jsou:
1. Charakterizovat možnosti a principy práce s daty WiFi sítí se zaměřením na Big Data.
2. Navrhnout vhodný postup zpracobaní a transformace dat.
3. Porovnat a vybrat vhodné algoritmy strojového učení pro kontinuální predikci

Metodika

Metodika řešení teoretické části bakalářské práce bude založena na studiu a analýze odborných informačních zdrojů.

Praktická část je založena na návrhu a realizaci systému pro predikovani obsazenosti prostor v areálů ČZU. Budou hledány vhodné algoritmy strojového učení pro kontinuální analýzu dat z WiFi přístupových bodů a následnou predikci obsazenosti.

Na základě syntézy teoretických poznatků a výsledků praktické časti budou formulovány závěry práce.

Doporučený rozsah práce

40 – 45 stran

Klíčová slova

big data, machine learning, hadoop, NOSQL, prediktivní modelování

Doporučené zdroje informací

HARRINGTON, Peter, c2012. Machine learnig in action. Shelter Island: Manning. ISBN 9781617290183.

HASTIE, Trevor J., Robert TIBSHIRANI a J. H. FRIEDMAN, c2009. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer. Springer series in statistics. ISBN 978-0-387-84857-0.

HENDL, J. *Přehled statistických metod : analýza a metaanalýza dat.* Praha: Portál, 2015. ISBN 978-80-262-0981-2.

SCHUTT, Rachel a Cathy O'NEIL, 2013. Doing Data Science: Straight Talk from the Frontline. Newton, Massachusetts, USA: O'Reilly Media. ISBN 978-1449358655.

WHITE, Tom, 2012. Hadoop: the definitive guide. 3rd ed. Sebastopol: O'Reilly. ISBN 978-1-449-31152-0.

Předběžný termín obhajoby

2022/23 ZS – PEF

Vedoucí práce

Ing. Jan Masner, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Konzultant

Ing. Vojtěch Novák

Elektronicky schváleno dne 27. 7. 2021

doc. Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 10. 2021

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 30. 10. 2022

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Prediktivní analýza obsazenosti prostor s pomocí dat z WiFi přístupových bodů" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15. března 2023 _____

Poděkování

Rád bych touto cestou poděkoval panu Janu Masneru za vstřícný přístup. Také bych rád poděkoval kolegovi Olegu Masailo a své rodině.

Prediktivní analýza obsazenosti prostor s pomocí dat z WiFi přístupových bodů

Abstrakt

Cílem dane práce je vytvoření modelu pro prognózu/predikci obsazenosti prostoru provozně ekonomické fakulty. Data pro tuhle práce budou dobyté pomoci skriptu, který bude napsán na jazyce Python a následného parsingu pomoci tohoto scriptu odkazů ve formátu HTML, odkazů, který budou získané z WiFi bodu.
Budou také charakterizovány možnosti a principy práce s daty v kontextu WiFi sítě s orientaci na "Big Data", v důsledku čeho, navrhněme vhodný způsob pro zpracování a transformaci těchto dat. Dále budou prozkoumání a porovnání mezi sebou různé algoritmy strojového učení s cílem odhalit lepší algoritmus pro naši cíl.

Klíčová slova: big data, machine learning, prediktivní modelování

Predictive analysis of space occupancy using data from WiFi access points

Abstract

The aim of the thesis is to develop a model for forecasting/predicting the occupancy of the Faculty of Economics and Management. The data for this thesis will be mined using a script that will be written on Python language and subsequent parsing using this script of links in HTML format, links that will be obtained from WiFi points.
The possibilities and principles of working with data in the context of a WiFi network will also be characterized with an orientation towards "Big Data", as a result, let us propose a suitable method for processing and transforming this data. Furthermore, different machine learning algorithms will be studied and compared with each other in order to discover a better algorithm for our target.

Keywords: big data, machine learning, predictive modeling

Seznam obrázků

Seznam tabulek

Odkazovaný seznam tabulek

1. Úvod

V dnešním světě obrovskou roli hrají tzv. “Big data” (Velké daty), v kontextu dnešních reálií lze tento fenomén pojmenovat novým “černým zlatem” nebolí “naftou”. Kolem dat točí se dnes vše – finanční trhy, banky, průmyslová výroba, mezinárodní korporaci a konglomeráty, státy a tak dále.

Člověk ve své každodenní činnosti, zanechává po sobě hodně “informačních stop”, buď to obyčejné použití bankovní karty, návštěva nějakých úřadu (tam on muže připojit se, například, na WiFi síť nebo jiným způsobem zanechat po sobě informační stopu), pohyb finančních prostředku, nákup nebo prodej nemovitosti, a dokonce když udělá zločin, v dnešní informační éru v dnešním globálním světě - co by neudělal, všude nechá po sobě informační stopu.

V dané práci bude provedena analýza obsazenosti prostor areálů ČZU (PEF) pomoci dat získaných z WiFi přístupových bodu a následná prediktivní analýza těchto dat s cílem odhalit zákonitosti a hypotézy obsazenosti budov a prostor provozně ekonomické fakulty. Data, která budou generovaný parsingem WiFi bodu – počet lidi v konkrétně zóně, v konkrétní moment času s určitými intervaly.

Motivaci pro tuto práci je osobní zájem a přání zkoumání tématu “Big Data”, prediktivní analytiky, strojového učení a následné aplikování v praxi, v tématu, který je cílem této práce. A také rozmyšlení na temat následovného použití prediktivní analýzy obsazenosti prostor v jiných oblastech vědy a reálného života.

V první, teoretické části práce, definujeme termín “Big Data” a prozkoumané různé algoritmy strojového učení pro odhalení optimálního pro použití v prediktivní analytice v rámci dané práce.

Druhá část bude zaměřena na realizaci praktické části a založena na znalostech získaných z teoretické části a programovacího jazyka “Python” s použitím knihoven, potřebných pro analýzu dat.

2. Cíl práce a metodika

2.1. Cíl práce

V této bakalářské práci klade za svoji hlavní cíl vytvořit prediktivní model pro analýzu s pomoci dat získaných s přístupových WiFi-bodu.

Pro realizaci dané cíle budeme potřebovat další:

1. Sběr aktuálních, relevantních a reprezentativních statistických dat.
2. Charakterizovat možnosti a principy práce se získanými daty a při tom budeme se opírat na principy práce s velkýma daty.
3. Navrhnout vhodnou posloupnost zpracování a transformaci velkých dat.
4. Porovnat a zvolit vhodně algoritmy strojového učení pro kontinuální predikci.

Předmětem výzkumu je nepřetržitě strojové učení s cílem zvýšení efektivity sestavení modelu.

Objektem výzkumu je data, které dostáváme z přístupových Wi-Fi bodu.

Použité metody a nářadí:
Pro zpracování dat bude využít programovací jazyk Python.

2.2. Metodika

Metodika řešení teoretické části bakalářské práce bude založena na studiu a analýze odborných informačních zdrojů.

Praktická část je založena na návrhu a realizaci systému pro predikovaní obsazenosti prostor v areálů ČZU. Budou hledány vhodné algoritmy strojového učení pro kontinuální analýzu dat z Wifi přístupových bodů a následnou predikci obsazenosti.

Na základě syntézy teoretických poznatků a výsledků praktické časti budou formulovány závěry práce.

3. Teoretická část práce

Tahle kapitola představuje teoretické podklady, charakteristiku možností a principů práce s velkýma daty, možnosti a principů které následovně budou použité v praktické části bakalářské práce.

3.1. Big data, co to je, jak to funguje, přehled současných technologií

3.1.1. Definice Big Data

Velká data představují soubor dat, který je příliš velký na to, aby mohl být zpracován tradičními metodami zpracování dat. V dnešním světě jsou Big Data jedním z nejvýznamnějších a nejdůležitějších pojmů, protože množství dat produkovaných a ukládaných společnostmi a organizacemi exponenciálně roste.

Big Data se využívají v různých oblastech včetně marketingu, vědy, medicíny, financí, výroby atd. Díky Big Data mohou analytici a odborníci identifikovat trendy, předpovídat události, vytvářet vzorce chování a rozhodovat na základě faktů.

Big Data tak zůstávají hlavním trendem dnešního světa a společnosti a organizace musí být připraveny tuto technologii využívat ke zlepšení svého fungování a udržení konkurenceschopnosti.

3.1.2. Nástroje pro zpracování velkých objemů dat

K ukládání velkých dat se obvykle používají několik různých nástrojů a technologií, z nichž některé jsou:

- Relační databáze: Relační databáze jsou tradičním způsobem ukládání dat, kde jsou data organizována do tabulek a relací. Tyto databáze jsou velmi robustní a podporují transakční zpracování a dotazování pomocí SQL.
- NoSQL databáze: NoSQL databáze jsou alternativou k relačním databázím a jsou obecně navrženy pro ukládání nestrukturovaných a velkých dat. Tyto databáze jsou často distribuované a mohou být horizontálně škálovatelné pro podporu velkého množství dat.

Budeme však v naší práci používat klasickou SQL relační databázi, abychom mohli shromažďovat naše data, protože nemáme cluster a naše data nejsou Big Data.

3.1.3. Základní principy práce s velkými daty

Práce s velkými daty je proces, při kterém se zpracovávají a analyzují obrovské objemy dat, které se obvykle nedají zpracovat tradičními metodami. Zde jsou základní principy práce s velkými daty:
Skládá se z více etap: Práce s velkými daty je často rozdělena na několik etap, jako je sběr dat, jejich ukládání, zpracování, analýza a interpretace výsledků. (1)

- Použití specializovaných technologií a nástrojů: Velké objemy dat nelze zpracovat běžnými metodami a technologiemi. Práce s velkými daty vyžaduje použití specializovaných technologií a nástrojů, jako jsou Hadoop, Spark, NoSQL databáze atd.
- Data musí být získána správně: Aby byla práce s velkými daty úspěšná, musí být data získána správně. To znamená, že data musí být relevantní, přesná a kompletní.
- Potřebujete správný tým: Práce s velkými daty vyžaduje tým lidí s různými zkušenostmi a dovednostmi. Tým musí být schopen řídit sběr dat, zpracování, analýzu a interpretaci výsledků.
- Bezpečnost dat je klíčová: Práce s velkými daty zahrnuje často citlivá data, takže je důležité zajistit jejich bezpečnost. To znamená, že data musí být šifrována, aby se zabránilo neoprávněnému přístupu.
- Kontrola kvality dat: Data musí být ověřena a kontrolována, aby se zajistila jejich kvalita. To znamená, že data musí být zkontrolována na přesnost, úplnost a relevanci.
- Práce s velkými daty je neustálý proces: Práce s velkými daty není jednorázový úkol. Je to neustálý proces, který vyžaduje neustálé aktualizace a úpravy.
- Využití strojového učení a umělé inteligence: Práce s velkými daty často zahrnuje využití strojového učení a umělé inteligence. Tyto technologie mohou pomoci při zpracování a analýze dat a při vytváření prediktivních modelů. (1)

3.2. Wifi sítě – definice, charakteristika a analýza

3.2.1. Definice

WiFi (Wireless Fidelity) je bezdrátová komunikační technologie, která umožňuje zařízením, jako jsou počítače, chytré telefony, tablety a další zařízení, připojit se k internetu nebo jiné síti bez použití kabelů.

Sítě WiFi se obvykle vytvářejí pomocí bezdrátových směrovačů, které k přenosu dat mezi zařízeními používají rádiové vlny. Zařízení, která se chtějí k síti připojit, musí mít kompatibilní bezdrátovou kartu, aby mohla se směrovačem komunikovat.

Aby zařízení našlo dostupné sítě WiFi, musí být v dosahu bezdrátového signálu. Zařízení obvykle zobrazí seznam dostupných sítí, ke kterým se lze připojit, a požádá o zadání hesla pro připojení k zabezpečené síti. (2) (3) (4)

3.2.2. Charakteristika Wi-Fi.

Sítě IEEE 802.11 (Wi-Fi) zažívají v současné době v mnoha odvětvích rychlý růst. růst v mnoha průmyslových odvětvích. Bezdrátové lokální sítě s vysokou hustotou (WLAN) obecně označují sítě, jejichž kapacita, tj. počet klientských zařízení podporovaných na metr čtvereční plochy, přesahuje kapacitu "tradičních" architektur. Ve většině případů se za hranici považuje jedno zařízení na metr čtvereční.

Síť Wi-Fi s vysokou hustotou označuje bezdrátové prostředí s vysokou koncentrací uživatelů, kde jsou uživatelé připojeni k bezdrátové síti a intenzivně využívají síťové služby. Může se jednat o sítě na místech, jako jsou konferenční sály, velké učebny, taneční parkety, stadiony, tisková centra, koncertní sály, letiště, obchodní haly, burzy, kasina atd. (3)

3.2.3. Analýza využívání sítí WiFi v kontextu velkých dat.

Stejně jako mnoho jiných technologií produkuje Wi-Fi velké množství dat, která lze využít k analýze a zlepšení výkonnosti sítě. Na druhou stranu velká data představují obrovské množství dat, jejichž zpracování vyžaduje velký výpočetní výkon. V případě Wi-Fi lze data získávat z různých zdrojů, jako jsou síťová zařízení, přístupové body atd.

Mezi hlavní možnosti velkých dat v kontextu sítě Wi-Fi patří – zpracování a analýza velkých dat za účelem optimalizace výkonu sítě a zlepšení kvality služeb. Vývoj inteligentních monitorovacích a řídicích systémů, které mohou automaticky analyzovat data a na základě výsledků přijímat rozhodnutí. Předvídání budoucích problémů v síti a

zavádění opatření k jejich předcházení. Analýza chování uživatelů a identifikace nejčastěji používaných zařízení a aplikací za účelem optimalizace sítě. Identifikace polohy uživatelů v rámci budov a využití těchto informací ke zlepšení výkonnosti sítě a optimalizaci služeb. (3)

3.2.4. Bezdrátová síť eduroam

Eduroam je bezdrátová síť určená pro vzdělávací a výzkumné organizace, které jsou členy mezinárodního společenství eduroam. Tato síť umožňuje uživatelům připojit se k bezdrátové síti v různých institucích a využívat přitom stejné přihlašovací údaje jako v jejich domovské organizaci. To znamená, že pokud jste studentem, akademickým pracovníkem nebo výzkumníkem na instituci, která je členem eduroam, můžete se připojit k bezdrátové síti eduroam na kterékoliv další instituci, která je také členem eduroam.

Existence sítě eduram znamená, že ji studenti mohou využívat zdarma, a proto máme velký počet uživatelů, kteří síť využívají denně.

Přihlášení k eduroam je zabezpečené pomocí šifrovaného protokolu EAP-TTLS nebo PEAP, který poskytuje vysokou úroveň bezpečnosti.

Připojení k eduroam nám umožní přístup k internetu a dalším službám, jako jsou e-maily, kalendáře a další vzdělávací a výzkumné zdroje. Síť eduroam je k dispozici v mnoha zemích po celém světě a její používání je zdarma pro uživatele, kteří jsou členy eduroam. (5)

3.3. Využitý software

K napsání všech skriptů pro sběr a zpracování dat byl použit programovací jazyk python a jeho četné moduly.

3.3.1. Python

Výhodou jazyka Python je jeho otevřenost. Software samotný je zdarma a jedná se o tzv. svobodný software (licence GNU GPL). (6)
Python je populární programovací jazyk, který má mnoho výhod. Některé z nejvýznamnějších výhod Pythonu jsou:

- Jednoduchost: Python má jednoduchou a intuitivní syntaxi, která usnadňuje psaní kódu a zvyšuje produktivitu.
- Velká komunita: Python má obrovskou komunitu vývojářů a uživatelů, kteří vytvářejí mnoho knihoven, balíčků a nástrojů, které jsou k dispozici zdarma pro ostatní uživatele.
- Multiplatformnost: Python běží na všech hlavních operačních systémech, včetně Windows, macOS a Linux, což z něj dělá velmi univerzální jazyk.
- Flexibilita: Python umožňuje programovat v mnoha různých stylech a paradigmách, včetně procedurálního, objektově orientovaného a funkcionálního programování.
- Široká škála aplikací: Python se používá v mnoha odvětvích, včetně web developmentu, data science, strojového učení, automatizace a mnoha dalších oblastech.
- Snadná instalace a použití: Python je snadno instalovatelný a použitelný, což z něj dělá vhodnou volbu pro začátečníky.
- Velmi dobře dokumentovaný: Python má velmi dobrou dokumentaci, což znamená, že je snadné najít informace o funkcích a knihovnách Pythonu.

Celkově lze říci, že Python je velmi výkonný a flexibilní programovací jazyk, který je snadno použitelný pro mnoho různých úkolů a aplikací. (6)

3.3.2. Moduly

Níže popisujeme moduly, které byly při práci použity ke sběru a zpracování dat.

Název	Popis funkcionality
urllib3	Klient HTTP
SQLAlchemy	SQLAlchemy je sada nástrojů SQL v jazyce Python a objektově relační mapovač, který vývojářům aplikací poskytuje plnou sílu a flexibilitu jazyka SQL.
Pandas	Pandas je rychlý, výkonný, flexibilní a snadno použitelný open source nástroj pro analýzu a manipulaci s daty.
beautifulsoup4	Beautiful Soup je knihovna, která usnadňuje získávání informací z webových stránek.

numpy	Knihovna poskytuje podporu vícerozměrných polí; podporu vysokoúrovňových matematických funkcí určených pro práci s vícerozměrnými poli.
statsmodels	je modul který poskytuje třídy a funkce pro odhadování mnoha různých statistických modelů, provádění statistických testů a zkoumání statistických dat.
matplotlib	je komplexní knihovna pro vytváření statických, animovaných a interaktivních vizualizací v jazyce Python.
sklearn	je knihovna určená pro strojové učení
prophet	implementuje model Prophet

3.3.3. Jupyter Notebook

Jupyter Notebook je interaktivní vývojové prostředí pro Python a další programovací jazyky, které umožňuje vytvářet a sdílet dokumenty obsahující kód, popisky, grafy a další multimediální prvky. Tento software se běžně používá pro vědecké výzkumy, datovou analýzu a výuku programování.

Mezi hlavní výhody Jupyter Notebook patří:

- Interaktivita: Umožňuje uživatelům snadno testovat a měnit kód a vidět výsledky interaktivně, což znamená, že mohou rychleji experimentovat s různými přístupy a způsoby řešení problémů.
- Snadné sdílení: Soubory Jupyter Notebook lze snadno sdílet a otevřít v různých formátech, včetně HTML, PDF nebo dokonce jako samostatný web.
- Podpora pro multimédia: Umožňuje vkládání obrázků, videí a zvukových souborů, což z něj činí vynikající nástroj pro vytváření dokumentů s různými multimediálními prvky.
- Podpora pro spoustu programovacích jazyků: Jupyter Notebook podporuje více než 40 programovacích jazyků, což umožňuje uživatelům pracovat s různými jazyky v jednom dokumentu.
- Flexibilita: Jupyter Notebook lze použít pro různé účely, včetně datové analýzy, strojového učení, vědeckého výzkumu, výuky a dalších oblastí. (7)

3.4. Časové řady a predikce časových řad

Časové řady (nebo dynamické řady) - statistický materiál shromážděný v různých časových okamžicích o hodnotě některých parametrů (v nejjednodušším případě jednoho) zkoumaného procesu. Každá jednotka statistického materiálu se nazývá měření nebo počet. V časové řadě musí být u každého vzorku uveden čas měření nebo pořadové číslo měření. (8)

Časová řada je posloupnost pozorování nebo měření nějaké proměnné v různých časových okamžicích. Může to být jakákoli proměnná, která se v čase mění, např. cena akcií, teplota, počet prodejů atd. V našem případě jsou naše data také časovou řadou, protože je sbíráme v pětiminutových intervalech.

Stejně jako většina ostatních typů analýzy i analýza časových řad předpokládá, že data obsahují systematickou složku (obvykle zahrnující několik složek) a náhodný šum (chybu), což znesnadňuje odhalení pravidelných složek. Většina technik analýzy časových řad zahrnuje různé techniky filtrování šumu, které umožní zřetelněji vidět pravidelnou komponentu. Většina pravidelných komponent v časových řadách patří do dvou tříd: jsou to buď trendové, nebo sezónní komponenty. Trend je obecná systematická lineární nebo nelineární komponenta, která se může v čase měnit. Sezónní komponenta je periodicky se opakující komponenta. Oba tyto typy pravidelných složek jsou v řadě často přítomny současně. (9) (8) (10)

Prediktivní algoritmy časových řad se používají k předpovídání budoucích hodnot na základě historických dat. Tyto algoritmy jsou obvykle používány v oblastech, jako jsou ekonomie, finance, meteorologie, průmyslová výroba a doprava. (10)

Zde je několik kroků, jak se používají prediktivní algoritmy časových řad:

1. Shromáždění dat: Začínáme shromážděním historických dat o časové řadě, kterou chceme předpovídat.
2. Vizualizace dat: Poté vizualizujeme data a analyzujeme trendy, sezónnost, cykly a další charakteristiky dat.
3. Výběr modelu: Na základě analýzy dat zvolíme model predikce, který se nejlépe hodí k našim datům a předpokládaným cílům.
4. Trénování modelu: Použijeme historická data pro trénování modelu predikce.

5. Testování modelu: Testujeme výkonnost modelu pomocí dat, která nebyla použita pro trénování, abychom zjistili, jak dobře náš model predikuje.
6. Vylepšení modelu: Na základě výsledků testování vylepšíme náš model predikce a opakujeme kroky 4 a 5, dokud nebudeme spokojeni s výsledky.
7. Předpovídání budoucích hodnot: Nakonec použijeme náš model k předpovídání budoucích hodnot na základě nových dat.
8. Příkladem prediktivního algoritmu časových řad může být například ARIMA (Autoregressive Integrated Moving Average), který se často používá v ekonomice a finanční analýze.

3.4.1. Průzkumná analýza časových řad

Průzkumná analýza časových řad je proces, kdy se prozkoumávají a analyzují data z časových řad s cílem získat užitečné informace o trendech, sezónnosti, cyklech a náhodných fluktuacích v datech. Níže jsou uvedeny některé základní kroky průzkumné analýzy časových řad:

- Vizualizace dat – Prvním krokem při průzkumné analýze časových řad je vizualizace dat. To umožňuje získat představu o trendech, sezónnosti a náhodných fluktuacích v datech. Vizualizace se může provádět pomocí grafů jako jsou časové řady, časové ploty, sezónní podprůměry a podobně.
- Detekce trendů – Trendy jsou dlouhodobé změny v datech a jsou obvykle důležité pro predikci budoucích hodnot. Trendy lze detekovat pomocí vizualizace dat nebo pomocí metod jako je lineární regrese.
- Detekce sezónnosti – Sezónnost se vyskytuje, když se opakují cykly v datech v pravidelných intervalech, obvykle během jednoho roku. Sezónnost lze detekovat pomocí sezónních podprůměrů nebo pomocí Fourierovy analýzy.
- Detekce cyklů – Cykly jsou dlouhodobé fluktuace v datech a lze je detekovat pomocí vizualizace dat nebo pomocí metod jako je spektrální analýza.
- Detekce náhodných fluktuací – Náhodné fluktuace jsou nevysvětlitelné změny v datech, které nelze spojit s trendem, sezónností ani cykly. Tyto fluktuace lze detekovat pomocí statistických metod jako je analýza autokorelace.

- Statistické analýzy – Kromě vizualizace dat lze provést také různé statistické analýzy, jako je například výpočet průměru, rozptylu, korelační analýza, regresní analýza, spektrální analýza a další.

Celkově je průzkumná analýza časových řad důležitým nástrojem pro porozumění datům v časových řadách a pro přípravu dat pro predikci budoucích hodnot. (10)

3.4.2. Faktory, které ovlivňují předpovídání časových řad

Předpovídání časových řad je proces, při kterém se na základě historických dat snažíme odhadnout, jak se bude daná časová řada vyvíjet v budoucnu. Existuje mnoho faktorů, které mohou ovlivnit přesnost předpovědi časové řady. Níže jsou uvedeny některé z hlavních faktorů:

- Trendy: Trendy jsou dlouhodobé změny v časové řadě, které mohou být způsobeny mnoha faktory, jako je například demografický vývoj, změny v technologii nebo v ekonomických podmínkách. Pokud trend přetrvává, může mít výrazný vliv na předpovědi časové řady.
- Sezónnost: Mnoho časových řad vykazuje pravidelné sezónní změny v průběhu roku. Například prodej zmrzliny bude pravděpodobně vyšší v létě než v zimě. Pokud je sezónnost silná, je důležité ji zahrnout do modelu předpovědi.
- Náhodné fluktuace: Časové řady mohou mít také náhodné fluktuace, které nemají žádný zjevný důvod. Tyto fluktuace se mohou vyskytovat v krátkodobém i dlouhodobém horizontu.
- Dataová kvalita: Předpovídání časových řad závisí na kvalitě dat. Pokud jsou data chybná nebo neúplná, mohou mít vliv na přesnost předpovědi.
- Velikost a četnost vzorků: Velikost a četnost vzorků mohou ovlivnit přesnost předpovědi časové řady. Pokud je vzorek příliš malý, může být předpověď nepřesná. Stejně tak, pokud je vzorek příliš velký, může být předpověď zbytečně složitá.
- Volba modelu: Existuje mnoho různých modelů, které mohou být použity pro předpovídání časových řad. Každý model má své vlastní výhody a nevýhody. Správná volba modelu může zlepšit přesnost předpovědi. (10)

3.4.3. Stacionarita

Než přejdeme k modelování, je důležitou vlastností časových řad stacionarita. Stacionarita je schopnost procesu neměnit své statistické charakteristiky v čase, tj. stálost očekávání, stálost rozptylu (homoskedasticita) a nezávislost kovarianční funkce na čase (měla by záviset pouze na vzdálenosti mezi pozorováními).

Proč je stacionarita tak důležitá? Ze stacionární řady lze snadno předpovídat, protože předpokládáme, že její budoucí statistické charakteristiky se nebudou lišit od pozorovaných současných charakteristik. Většina modelů časových řad tyto charakteristiky (očekávání a rozptyl) tak či onak modeluje a předpovídá, takže pokud je původní řada nestacionární, ukáže se, že předpovědi jsou nesprávné. Bohužel většina časových řad, se kterými se setkáme mimo tréninkový materiál, je nestacionární, ale s tím se lze (a mělo by se) vypořádat. (10)

3.4.4. Algoritmy strojového učení pro kontinuální predikci

Algoritmy strojového učení pro kontinuální predikci jsou používány pro předpovídání hodnoty nebo trendu proměnné v čase. Tyto algoritmy jsou často používány v oblastech jako jsou finance, zdravotnictví a průmysl.

Zde je několik příkladů algoritmů strojového učení pro kontinuální predikci:

- Lineární regrese: Lineární regrese se používá k predikci kontinuálních proměnných na základě lineárního vztahu mezi vstupními a výstupními proměnnými. Tento algoritmus se používá v situacích, kdy se očekává, že změna v jedné proměnné bude mít lineární dopad na druhou proměnnou.
- ARIMA: ARIMA (AutoRegressive Integrated Moving Average) je statistický model pro predikci časových řad, který se skládá z autoregresivního (AR) modelu, integrovaného (I) modelu a krokového průměru (MA) modelu. ARIMA se používá k predikci vývoje časových řad.

- Exponenciální vyhlazování: Exponenciální vyhlazování je algoritmus, který se používá k predikci časových řad, které mají trend a sezónnost. Tento algoritmus se používá k predikci krátkodobých výkyvů a změn v časových řadách.

- LSTM: LSTM (Long Short-Term Memory) je druh rekurentní neuronové sítě, která se používá k predikci časových řad a sekvenčních dat. Tento algoritmus je schopen zachytit dlouhodobé závislosti v časových řadách a je často používán v oblastech jako jsou předpověď poptávky, prognózování trhu a předpovídání finančních ukazatelů.
- Gradient Boosting: Gradient Boosting je technika, která se používá k predikci kontinuálních proměnných v rámci souboru trénovacích dat. Tento algoritmus se používá k vytvoření souboru rozhodovacích stromů, které jsou následně použity k predikci nových dat. Gradient Boosting se často používá v oblastech jako jsou finanční prognózování a analýza dat.

Tyto algoritmy mohou být kombinovány a upravovány v závislosti na specifických požadavcích a charakteristikách datových souborů. (10)

3.4.5. Algoritmy, které budeme porovnávat

3.4.6. Prophet

Prophet je open source knihovna vytvořená Facebookem pro predikci časových řad. Tento model kombinuje několik funkcionalit pro analýzu časových řad, jako jsou sezónnost, trend a vliv prázdnin. Prophet také umí pracovat s nepravidelnými intervaly dat a s chybějícími hodnotami.

Řekněme si něco o tom, jak knihovna Prophet funguje. Jedná se v podstatě o aditivní regresní model, který se skládá z následujících komponent:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

1. **Sezónní komponenty** s(t) jsou zodpovědné za modelování periodických změn spojených s týdenní a roční sezónností. Týdenní sezónnost se modeluje pomocí `dummy variables`. Přidává se šest dalších funkcí, např, `[monday, tuesday, wednesday, thursday, friday, saturday]` které nabývají hodnot 0 a 1 v závislosti na datu. Komponenta `Sunday` odpovídající sedmému dni v týdnu se nepřidává, protože bude lineárně záviset na ostatních dnech v týdnu, což by ovlivnilo model. Roční sezónnost je modelována Fourierovou řadou.

2. **Trend** g(t) - je logistická funkce. Logistická funkce má tvar:

$$g(t) = \frac{C}{1 + exp(-k(t-b))}$$ umožňuje modelovat růst s nasycením, kdy se rychlost růstu snižuje s rostoucím ukazatelem. Typickým příkladem je růst počtu uživatelů aplikace nebo webové stránky.
3. **Komponenta** h(t) - odpovídá za uživatelem definované abnormální dny, včetně nepravidelných dnů, jako jsou svátky nebo výjimečné události.
4. **Chyba** є (t) - obsahuje informace, které model nebere v úvahu.
   (10) (11)

Hlavní výhody Python Prophet jsou:

- Snadné použití: Prophet poskytuje jednoduché API pro vytváření modelů předpovědí časových řad.
- Flexibilita: Prophet umožňuje uživateli přizpůsobit mnoho parametrů modelu, například typ trendu, typ sezónnosti a regresní parametry.
- Škálovatelnost: Prophet si poradí s velkými soubory dat s tisíci časových řad.
- Podpora zpracování odlehlých hodnot a chybějících hodnot: Prophet dokáže pracovat s časovými řadami obsahujícími odlehlé hodnoty a chybějící hodnoty. (11)

3.4.6.1 Přesnost algoritmu Prophet

Přesnost algoritmu Prophet závisí na mnoha faktorech, jako je kvalita dat, délka časové řady, množství dostupných informací a složitost modelu. V některých případech může být Prophet velmi přesný a v jiných případech může být méně přesný. Proto je důležité důkladně vyhodnotit výsledky a zvážit, zda jsou dostatečně přesné pro konkrétní účel.

V praxi je však Prophet považován za velmi přesný algoritmus pro predikci časových řad, zejména pokud jsou k dispozici dostatečná množství dat a informací. (11)

3.4.7. SARIMA

Algoritmus SARIMA (Seasonal Autoregressive Integrated Moving Average) je jedním z nejrozšířenějších algoritmů pro predikci časových řad. Je založen na rozšíření algoritmu ARIMA (Autoregressive Integrated Moving Average) o sezónní komponentu.

ARIMA je algoritmus, který umožňuje modelovat časovou řadu pomocí kombinace autoregresních, integrovaných a pohyblivých průměrů. Autoregresní komponenta modeluje závislost řady na jejích minulých hodnotách, integrovaná komponenta zahrnuje rozdíly mezi po sobě jdoucími hodnotami pro vyrovnání náhodných fluktuací a pohyblivý průměr slouží k vyhlazení šumu. Nicméně, ARIMA není vhodný pro modelování sezónních trendů v časových řadách. (10)

SARIMA je tedy rozšířením ARIMA pro modelování sezónních trendů. Algoritmus umožňuje modelovat nejen samotnou sezónnost, ale také zahrnuje autoregresní, integrované a pohyblivé průměry. SARIMA se skládá ze čtyř komponent: autoregresní komponenty (AR), integrované komponenty (I), pohyblivého průměru (MA) a sezónní komponenty (S).

AR komponenta modeluje vliv minulých hodnot na aktuální hodnotu časové řady. I komponenta slouží k vyrovnání náhodných fluktuací pomocí diferencování. MA komponenta modeluje šum v časové řadě. Sezónní komponenta modeluje periodické změny v časové řadě.

3.4.7.1 Přesnost algoritmu SARIMA

Závisí na několika faktorech, jako jsou:

- Kvalita a množství dat: SARIMA může být použit pro analýzu časových řad s malým množstvím dat, ale přesnost modelu se zvyšuje s větším množstvím dat. Kvalita dat, jako je například přítomnost výrazných odlehlých hodnot nebo chybějících dat, může také ovlivnit přesnost algoritmu.

- Volba modelu SARIMA: Volba správného modelu SARIMA je klíčová pro dosažení vysoké přesnosti. Je důležité správně identifikovat řády AR, MA a rozdílové operátory pro přesné modelování časové řady.

- Přizpůsobení modelu: Některé SARIMA modely mohou být velmi přizpůsobivé a dosáhnout vysoké přesnosti, pokud jsou správně kalibrovány na základě historických dat. Avšak, přizpůsobení modelu může vést k nadměrnému přizpůsobení datům a model může mít horší výkonnost na nových datech.

- Validace modelu: Validace modelu je důležitá pro kontrolu přesnosti predikce. Obvyklým přístupem je použití cross-validace, kde jsou data rozdělena na trénovací a testovací sady. Model je trénován na trénovací sadě a přesnost predikce je měřena na testovací sadě.

V závislosti na těchto faktorech může být přesnost algoritmu SARIMA různá. Obecně je však SARIMA považován za účinný a přesný algoritmus pro predikci časových řad. (10)

4. Praktická část práce

V praktické části této práce jsme na shromážděná data aplikovali statistické metody a prognostické algoritmy. Data byla sesbírána za období patnácti měsíců. Obsah dat bude popsán, analyzován a prognózován níže.

4.1. Sběr a transformace dat - architektonické řešení - ETL Pipelina

ETL pipeline je proces, který se používá k integraci, přeměně a uložení dat z různých zdrojů do cílového úložiště, jako je například data warehouse nebo data lake. ETL je zkratka pro Extract, Transform, Load.

Extrahování (Extract) - Prvním krokem v ETL pipeline je extrakce dat z různých zdrojů, jako jsou soubory, relační databáze, NoSQL databáze, webové služby nebo cloudové úložiště. Tato data se obvykle ukládají do dočasného úložiště jako jsou například datové proudy nebo soubory.

Transformace (Transform) - Dalším krokem v ETL pipeline je přeměna dat. Během tohoto kroku jsou data upravována, filtrována, přetvářena a strukturována do formátu, který odpovídá cílovému úložišti. Transformace mohou zahrnovat odstranění neplatných dat, sjednocení datových formátů a vytvoření nových výpočtů.

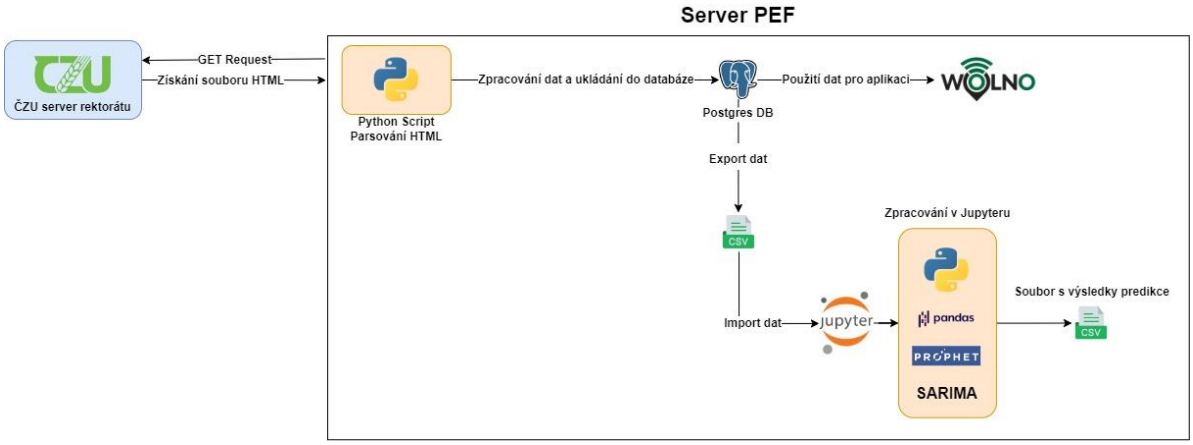
Nahrání (Load) - Posledním krokem v ETL pipeline je nahrání dat do cílového úložiště. Cílové úložiště může být relační nebo nerelační databáze, datové skladovací řešení, nebo obecně jakékoliv úložiště, které je vhodné pro daný účel. Nahrání dat do cílového úložiště může probíhat v reálném čase nebo v časových intervalech.

ETL pipeline je důležitý proces pro organizace, které potřebují shromažďovat a analyzovat data z různých zdrojů. Tento proces umožňuje organizacím efektivně získávat, přetvářet a ukládat data pro analýzu a využití v různých aplikacích a procesech.

4.1.1. Naše ETL Pipelina

Tento obrázek jasně ukazuje naše architektonické řešení. Nejprve pomocí pythonovského skriptu načteme ze serveru string HTML a poté jej zpracujeme. Vezmeme

hodnoty, které potřebujeme, a po konverzi je uložíme do naší databáze. Tato data se pak používají v aplikaci wolno k vizualizaci obsazenosti prostor fakulty v reálném čase. Pro naše řešení generujeme soubor csv z dat v databázi a poté tento soubor použijeme k analýze a predikci.



Obrázek 1: Architektonické řešení. Zdroj: Vlastní zpracování

4.1.2. Výzkumný datový soubor

Zdrojem našich dat jsou body wi-fi, ke kterým jsou studenti připojeni. Ke sběru dat nám katedra poskytla vlastní server, který je připojen k interní síti.

Pro sběr dat bylo vytvořeno následující schéma databázové tabulky:

Název sloupce	Typ	Detailní popis
id	Bigint Auto Increment (Primary key)	Unikátní číslo záznamu, které je zároveň automatickým inkrementem.
connUsers	Integer	Počet připojených uživatelů v daném čase v celých číslech
timemark	Timestamp	Čas sběru dat ve formátu YYYY-MM-DD HH:MM:SS
sectorId	Integer	Číslo zóny na mapě, ve které se může nacházet jeden nebo více adaptérů Wi-Fi.

4.1.3. Popis skriptu pro sběr dat a jeho fungování

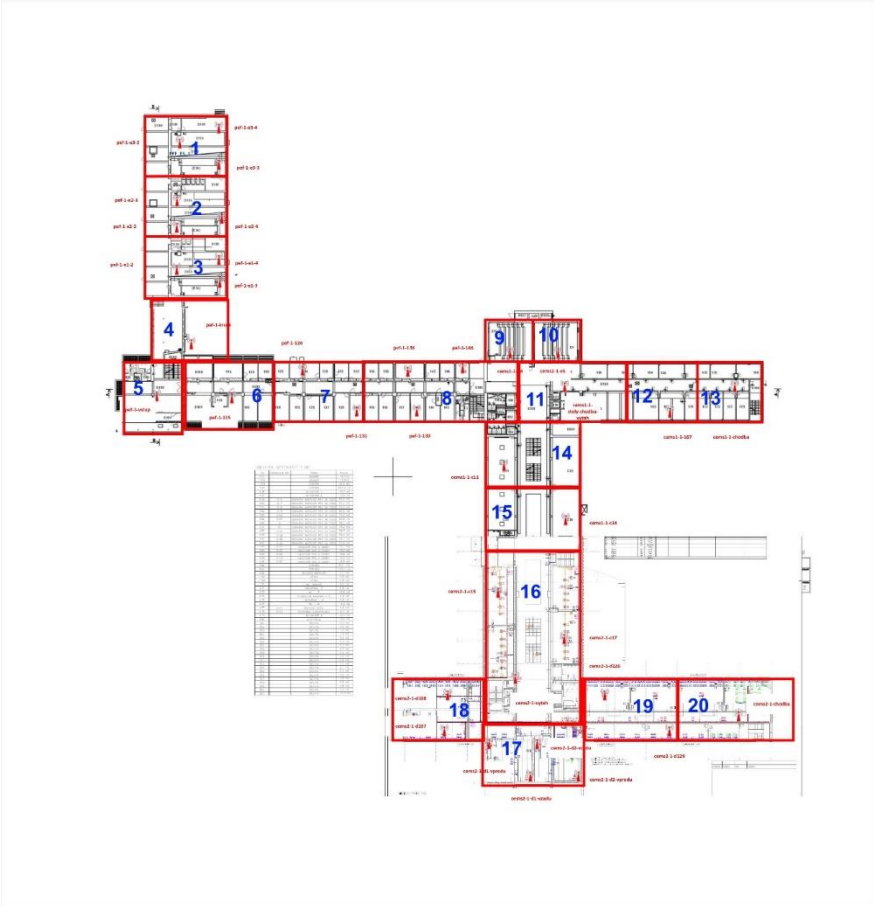
Skript běží na univerzitním serveru na Linuxu, pomocí crontabu jej spouštíme každých pět minut.

Sběr dat se skládá ze dvou souborů

1. Soubor wifi_parser.py obsahuje skript pro parsování dat z přístupových bodů WiFi a jejich ukládání do databáze.
2. Soubor ap_zones.py je soubor pro mapování (statické mapování) existujících přístupových bodů do zón, které máme na mapě (viz obrázky). Jedna zóna může obsahovat jeden nebo více přístupových bodů. ID zón závisí na indexu zóny v seznamu zón.

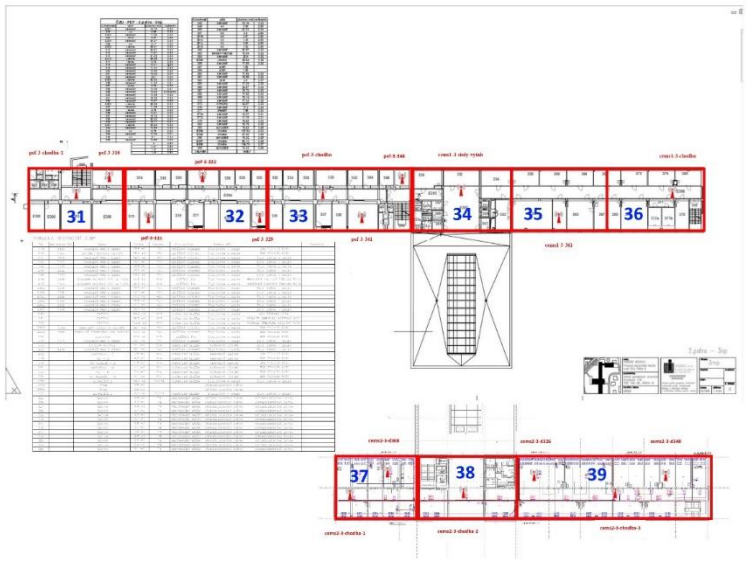
4.1.4. Plán fakulty a rozdělení bodů wi-fi do zón

Na pláncích níže vidíte, jak jsme rozdělili fakultu do zón, ze kterých budeme sbírat data:



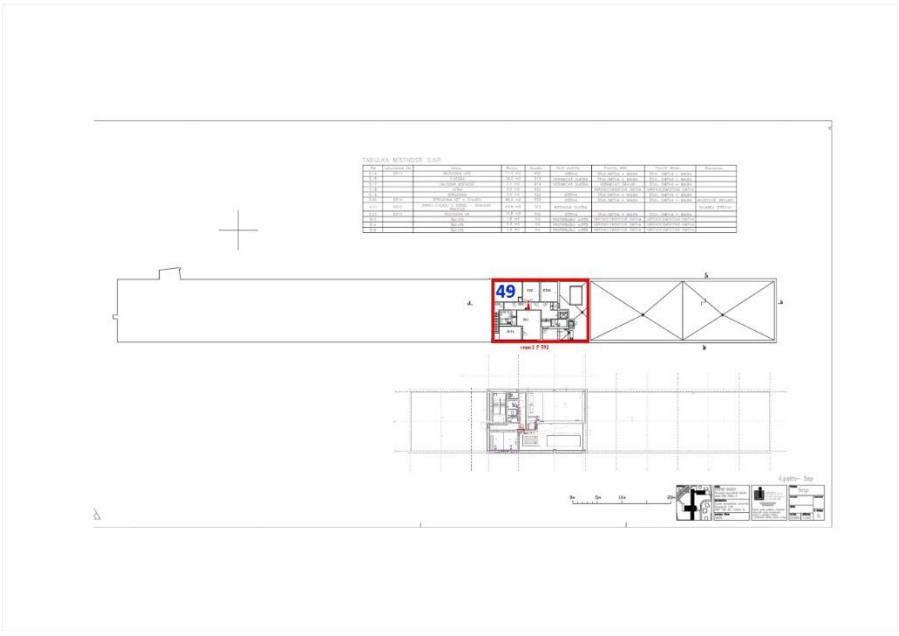
Obrázek 2: 1.NP. Zdroj: Vlastní zpracování

Obrázek 3: 2.NP. Zdroj: Vlastní zpracování



Obrázek 4: 3.NP. Zdroj: Vlastní zpracování

Obrázek 5: 5.NP. Zdroj: Vlastní zpracování



Obrázek 6:6.NP. Zdroj: Vlastní zpracování

4.1.5. Transformace datového souboru

Vzhledem k tomu, že náš soubor je poměrně velký a vyžaduje velký výpočetní výkon, a protože potřebujeme nesystematizovaná data transformovat do podoby, kterou lze snadno analyzovat, bylo provedeno následující:

1. Funkce wolno_df_processing sumarizuje všechny zóny v určitém čase (konkrétně každých pět minut) - tím získáte celkový počet unikátních uživatelů připojených k síti - výsledkem je datový soubor s globálním počtem unikátních uživatelů připojených v pětiminutových intervalech.
2. Dále hledáme klouzavý průměr pro každých dvacet čtyři hodin, tj. den. Výsledkem je průměrný počet uživatelů, kteří navštívili fakultu za celý den.
3. Výsledkem všech transformací je jediná časová řada pro predikci.

4.1.6. Zkoumání datového souboru

timemark	connUsers
2022-12-12	100.034722
2022-12-13	255.510417
2022-12-14	278.458333
2022-12-15	267.489583
2022-12-16	226.451389
...	...
2023-02-22	331.690972
2023-02-23	332.853659
2023-02-24	124.649306
2023-02-25	8.163194
2023-02-26	9.507317

Obrázek 7: Příklad dat. Zdroj: Vlastní zpracování

	connUsers
count	477.000000
mean	108.792540
std	123.196329
min	2.594262
25%	9.236111
50%	57.838028
75%	198.236111
max	1042.003472

Obrázek 8: Základní charakteristiky dat. Zdroj: Vlastní zpracování

Údaje "connUsers" popisují některé charakteristiky týkající se počtu uživatelů, kteří se připojují k naší síti. V tomto případě je uvedeno 477 měření této metriky.

První hodnota, "count", udává počet záznamů v datech, které tvoří vzorek pro analýzu. V tomto případě se jedná o 477 záznamů.

Hodnota "mean" je 108,792540. To znamená, že průměrný počet uživatelů přistupujících k systému v každém měření vzorku je přibližně 109. To může být užitečná informace pro určení průměrného zatížení systému, aby mohl adekvátně zpracovávat požadavky uživatelů.

Směrodatná odchylka ("std") je 123,196329. Jedná se o míru toho, jak moc se jednotlivá měření vzorku odchylují od průměru. V tomto případě je směrodatná odchylka poměrně vysoká, což může znamenat, že existují velké rozdíly v počtu uživatelů, kteří se k systému připojují. To může být užitečná informace pro určení optimální infrastruktury, která se dokáže vyrovnat s kolísáním počtu uživatelů.

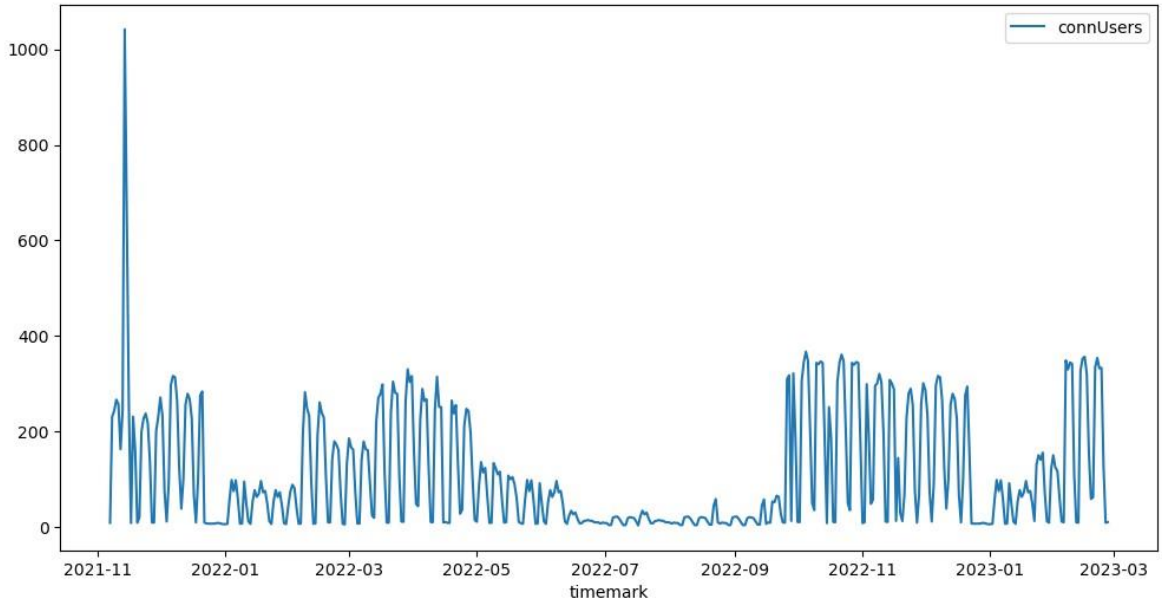
Minimální hodnota ("min") je 2,594262. To znamená, že nejmenší počet uživatelů, kteří se připojili k systému ve vzorku, byl 2. To může pomoci při určování minimálního zatížení systému.

První kvartil ("25%") je 9,236111. Jedná se o údaj, který udává počet uživatelů, kteří se k systému připojili ve 25 % měření ve vzorku. To znamená, že 25 % měření má počet připojených uživatelů menší nebo roven 9.

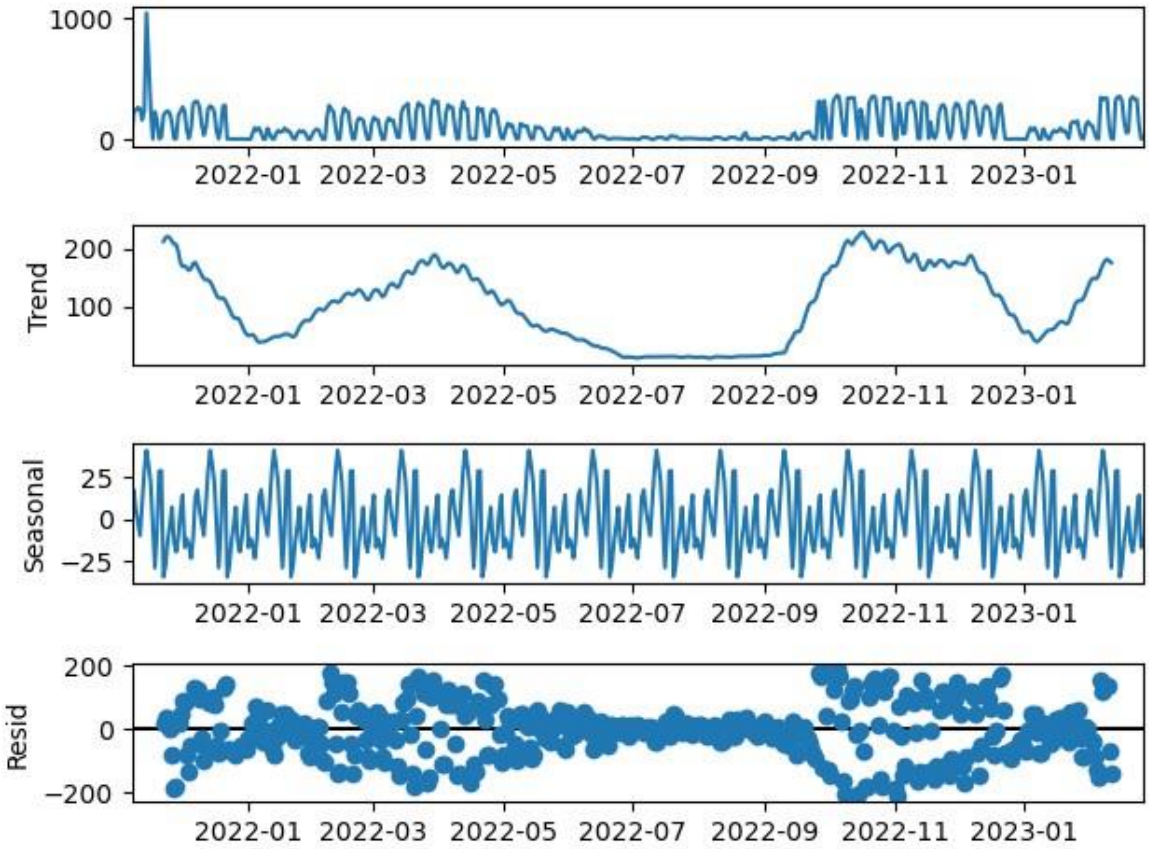
Mediánová hodnota ("50 %") je 57,838028. To znamená, že 50 % měření má počet připojených uživatelů menší nebo roven 58.

Třetí kvartil ("75 %") je 198,236111. To je hodnota, která udává počet uživatelů, kteří byli připojeni v 75 % měření vzorku. To znamená, že 75 % měření má počet připojených uživatelů menší nebo roven 198.

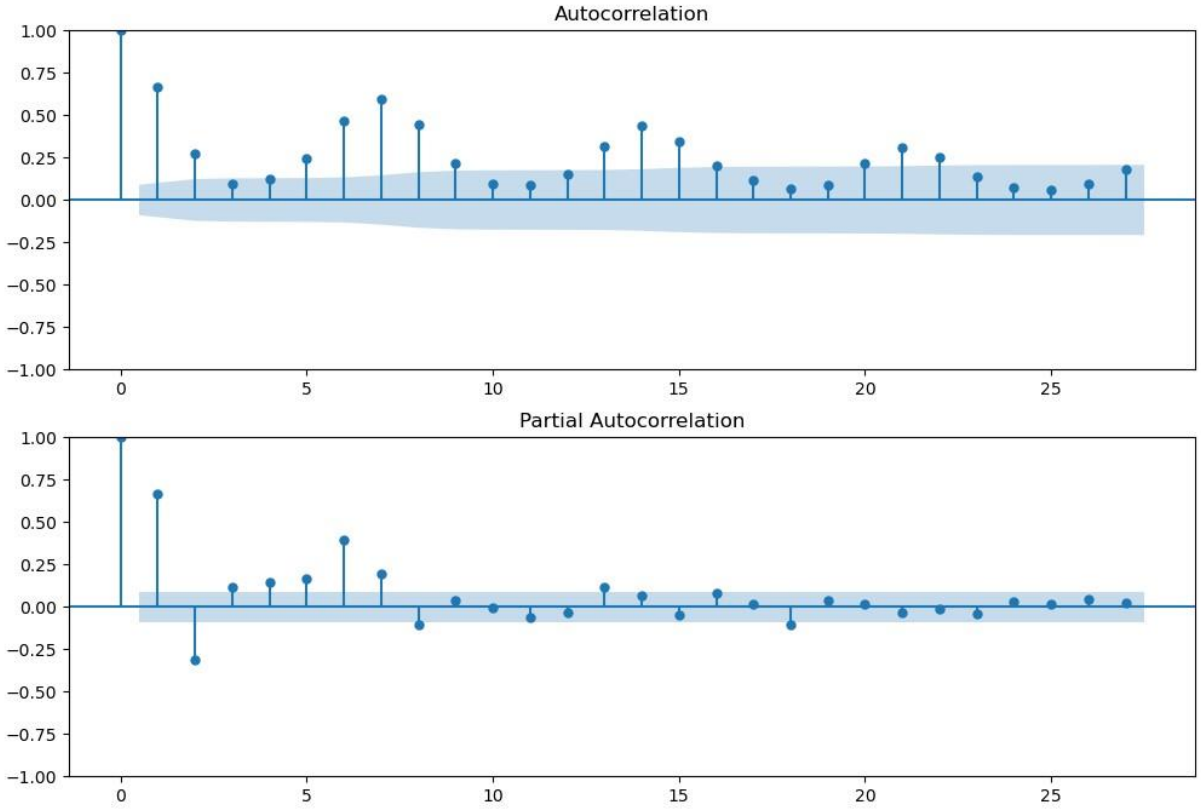
Maximální hodnota ("max") je 1042,003472.



Obrázek 9: Graf časové řady. Zdroj: Vlastní zpracování



Obrázek 10: Grafy komponent časových řad. Zdroj: Vlastní zpracování



Obrázek 11: Korelogramy ACF a PACF. Zdroj: Vlastní zpracování

Po prozkoumání korelogramu PACF lze dojít k závěru, že p = 1, protože pouze 1 zpoždění se významně liší od nuly. Z korelogramu ACF lze vyvodit, že q = 1, protože po 1. zpoždění hodnoty funkce prudce klesají.

Z grafů je patrná určitá sezónnost a trend.

Abychom otestovali stacionaritu, provedeme Dickey-Fullerův test na přítomnost jednotkových kořenů.:

```
def ad_test(dataset):
    dftest = adfuller(dataset, autolag = 'AIC')
    print(f'1. ADF : {dftest[0]}')
    print(f'2. P-Value : {dftest[1]}')
    print(f'3. Num Of Lags : {dftest[2]}')
    print(f'4. Num Of Observations Used For ADF Regression and Critical Values Calculation :
{dftest[3]}')
    print(f'5. Critical Values :')
    for key, val in dftest[4].items():
        print('\t', key, ': ', val)
    if dftest[0]> dftest[4]['5%']:
        print('existují jednotkové kořeny, řada není stacionární')
    else:
        print('neexistují jednotkové kořeny, řada je stacionární')
```

Output:

```
1. ADF : -2.6765224352311714
2. P-Value : 0.0781897667496434
3. Num Of Lags : 17
4. Num Of Observations Used For ADF Regression and Critical Values Calculation : 459
5. Critical Values :
        1% :  -3.4446773373329576
        5% :  -2.8678574606780654
        10% :  -2.5701349669405404
existují jednotkové kořeny, řada není stacionární
```

Test potvrdil předpoklady o nestacionaritě časových řad. V mnoha případech nám to umožňuje vzít diferenci řady. Pokud jsou například první diference řady stacionární, pak se nazývá integrovaná řada prvního řádu.
Určeme tedy řád integrované řady pro naši řadu:

```
df2 = df.diff(periods=1).dropna()
```

Ve výše uvedeném kódu vypočítá funkce diff() rozdíl původního řádku a řádku s daným posunem periody. Posunutí periody se předává jako parametr period. Protože první hodnota v rozdílu bude neurčitá, musíme se jí zbavit a použít k tomuto účelu metodu dropna().
Zkontrolujme výslednou řadu na stacionaritu:

1. ADF : -6.484169463979732
2. P-Value : 1.2730693408703131e-08
3. Num Of Lags : 17
4. Num Of Observations Used For ADF Regression and Critical Values Calculation : 458
5. Critical Values :
    1% : -3.4447087976702284
    5% : -2.867871300049488
    10% : -2.5701423432047443

neexistují jednotkové kořeny, řada je stacionární
Jak je patrné z výše uvedeného kódu, výsledná řada prvních diferencí je téměř stacionární. Chceme-li se o tom přesvědčit, rozdělime ji na několik intervalů a zkontrolujeme matematické očekávání v různých intervalech:

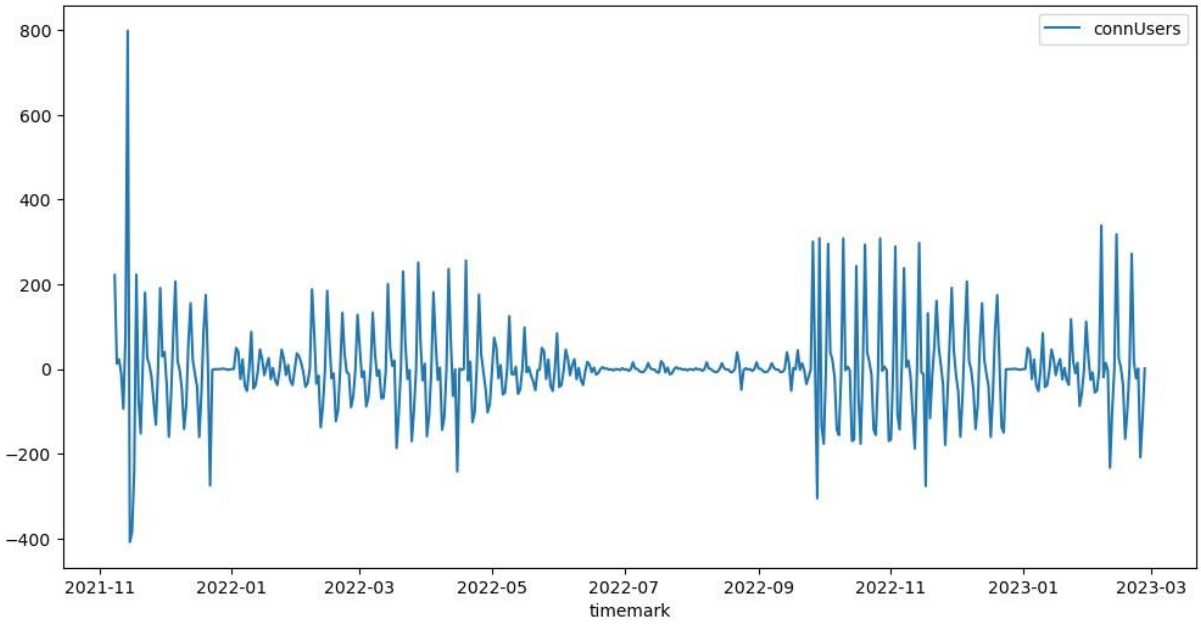
```
m = df2.index[len(df2.index)/2+1]
r1 = sm.stats.DescrStatsW(df2[m:])
r2 = sm.stats.DescrStatsW(df2[:m])
print('p-value: ', sm.stats.CompareMeans(r1,r2).ttest_ind()[1])
```

Output:

```
p-value:  [0.99822489]
```

Vysoká p-hodnota nám umožňuje tvrdit, že nulová hypotéza o rovnosti středních hodnot je správná, což znamená, že řada je stacionární.

Zbývá zjistit, zda neexistuje žádný trend, a proto zobrazme naši novou časovou řadu:



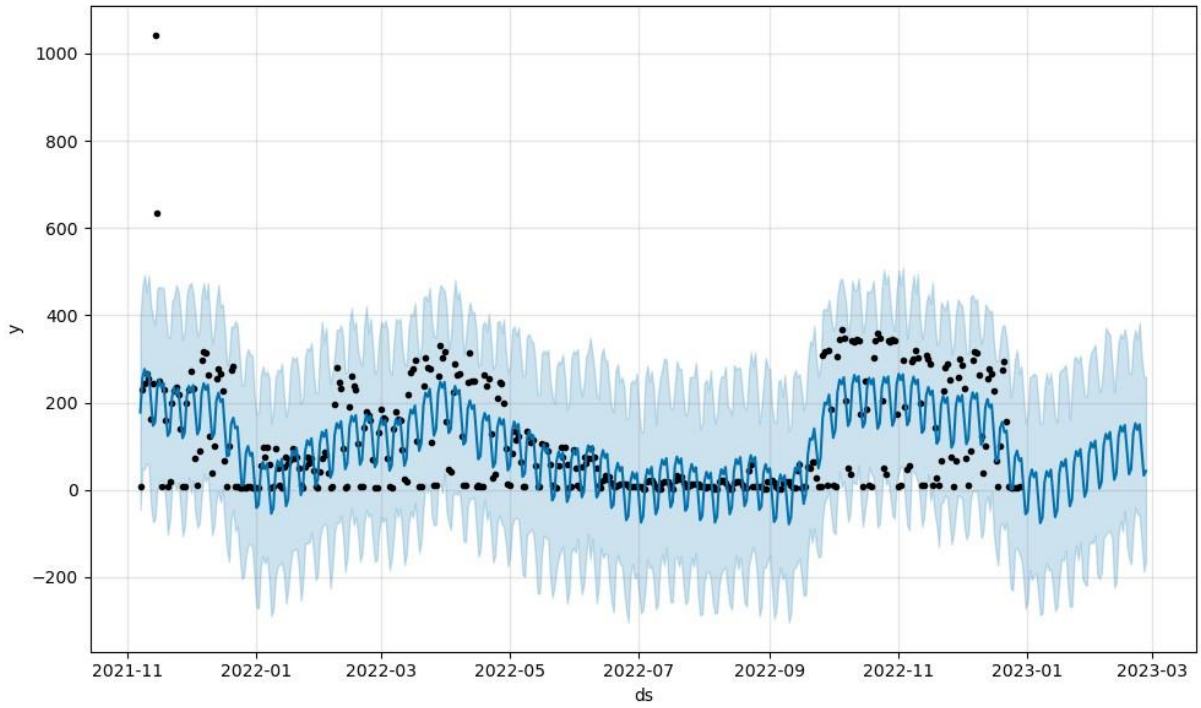
Obrázek 12: Nová časová řada. Zdroj: Vlastní zpracování

Trend skutečně neexistuje, takže řada prvních diferencí je stacionární a naše původní řada je integrovanou řadou prvního řádu.

4.2. Predikční model Prophet

Tento model nám umožňuje velmi snadno vytvořit obecnou předpověď a vizualizovat ji:

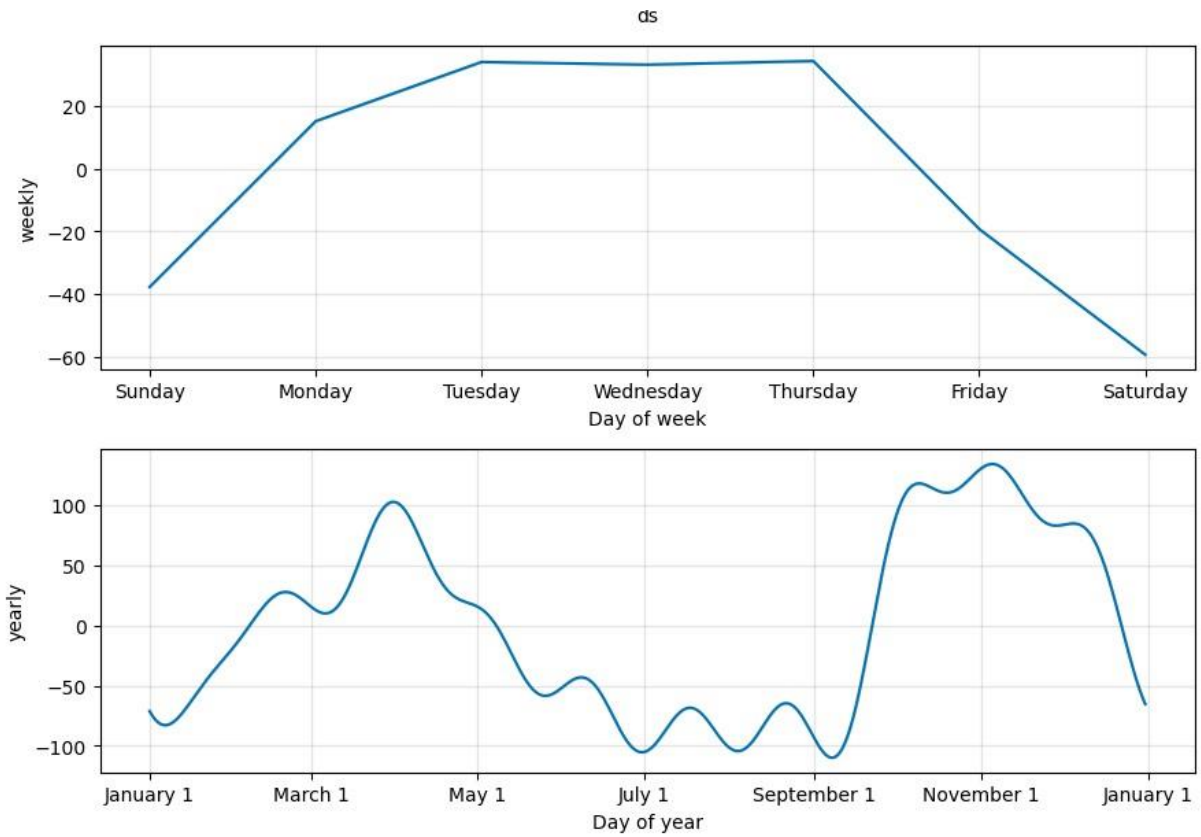
```
from prophet import Prophet
predictions = 60

df = df.reset_index()
df.columns = ['ds', 'y']

train_df = df[:-60]

m = Prophet(growth='linear',
            changepoints=None,
            n_changepoints=100, # 25 before
            changepoint_range=0.8,
            yearly_seasonality=True,
            weekly_seasonality=True,
            daily_seasonality=False,
            holidays=None,
            seasonality_mode='additive',
            seasonality_prior_scale=10.0,
            holidays_prior_scale=10.0,
            changepoint_prior_scale=0.05,
            mcmc_samples=0,
            interval_width=0.99,
            uncertainty_samples=1000,
            stan_backend=None
           )

m.fit(train_df)

future = m.make_future_dataframe(periods=predictions)
forecast = m.predict(future)

m.plot(forecast);
```



Obrázek 13: Predikce časových řad – Prophet. Zdroj: Vlastní zpracování

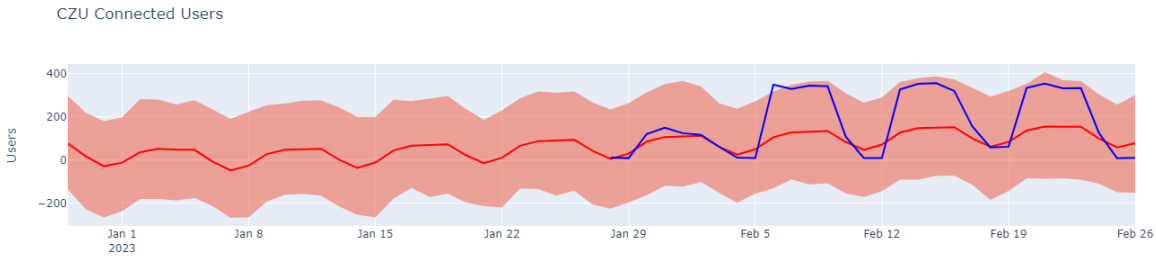
Můžeme také znázornit týdenní a roční trendy, které algoritmus určuje:

m.plot_components(forecast);



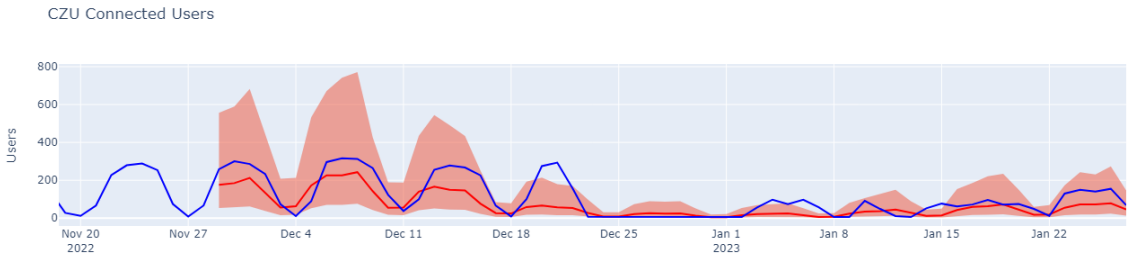
Obrázek 14: Týdenní a roční trendy – Prophet. Zdroj: Vlastní zpracování

Nakonec vizualizujeme naši předpověď:



Obrázek 15: Primární prognóza. Zdroj: Vlastní zpracování

Po transformaci Box-Cox zvýší se kvalita modelu a dostaneme:



Obrázek 16: Prognóza po transformaci Box-Cox. Zdroj: Vlastní zpracování

4.3. Predikční model SARIMA

4.3.1. hledání optimálních hodnot SARIMA

Pro predikční model sarima, musíme najít optimální hodnoty pro náš model.

Za tímto účelem jsme vytvořili následující funkci:

```
def optimize_SARIMA(parameters_list, d, D, s, exog):
    """
        Return dataframe with parameters, corresponding AIC and SSE

        parameters_list - list with (p, q, P, Q) tuples
        d - integration order
        D - seasonal integration order
        s - length of season
        exog - the exogenous variable
    """

    results = []

    for param in tqdm_notebook(parameters_list):
        try:
```

```
            model = SARIMAX(exog, order=(param[0], d, param[1]), seasonal_order=(param[2],
D, param[3], s)).fit(disp=-1)
        except:
            continue

        aic = model.aic
        results.append([param, aic])

    result_df = pd.DataFrame(results)
    result_df.columns = ['(p,q)x(P,Q)', 'AIC']
    #Sort in ascending order, lower AIC is better
    result_df = result_df.sort_values(by='AIC', ascending=True).reset_index(drop=True)

    return result_df
```

Nalezení optimálních hodnot modelu:

```
p = range(0, 4, 1)
d = 1
q = range(0, 4, 1)
P = range(0, 4, 1)
D = 1
Q = range(0, 4, 1)
s = 4
parameters = product(p, q, P, Q)
parameters_list = list(parameters)
print(len(parameters_list))


result_df = optimize_SARIMA(parameters_list, 1, 1, 4, df['connUsers'])
```

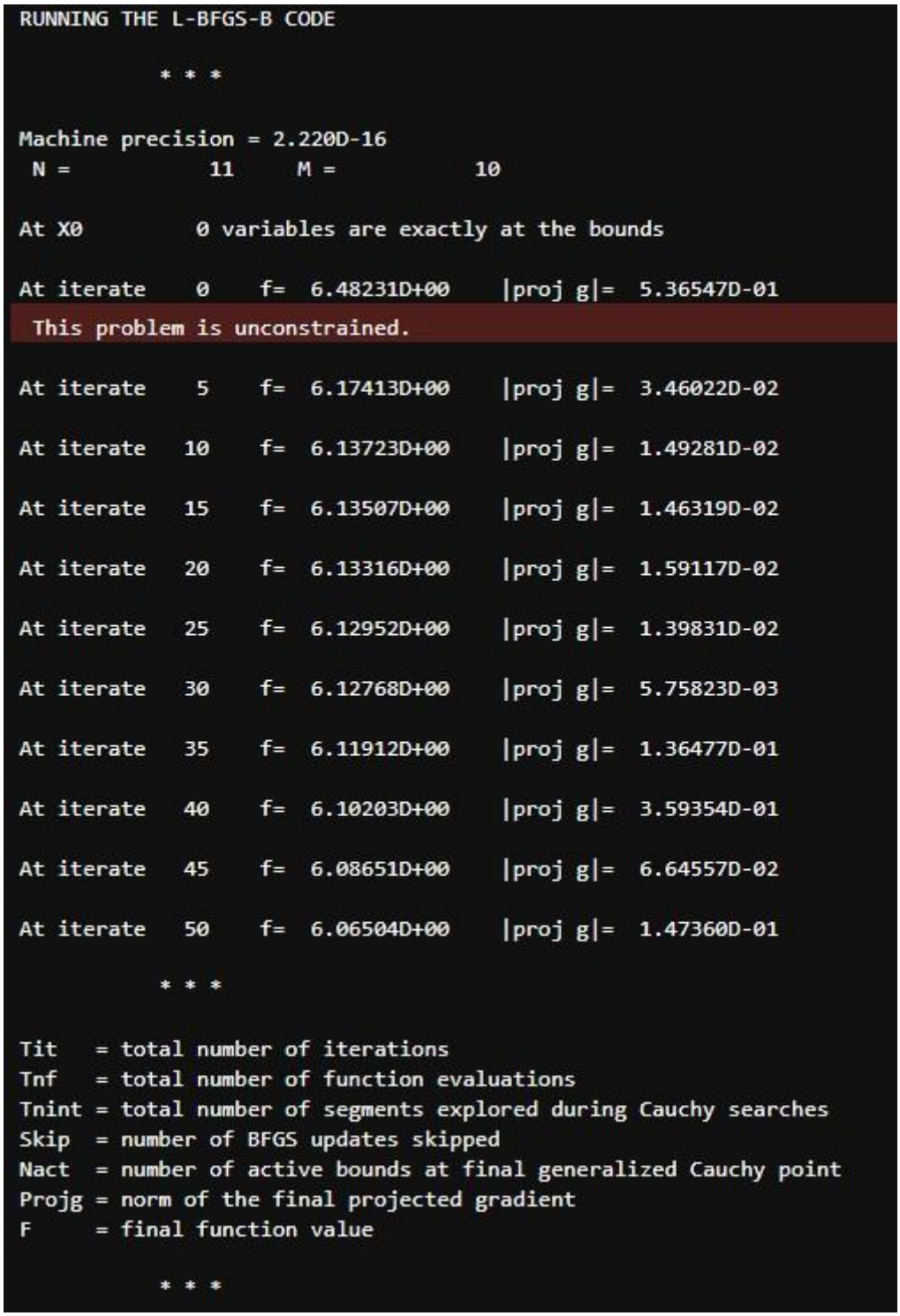
Po analýze časové řady získáme optimální hodnoty:

```
# (2, 2, 3) (2, 2, 3, 4)  9658.620710
```

Vložíme je do našeho modelu:

```
best_model = SARIMAX(df['connUsers'], order=(2, 2, 3), seasonal_order=(2, 2, 3, 4),
enforce_stationarity=False).fit(dis=-1)
```

Output:

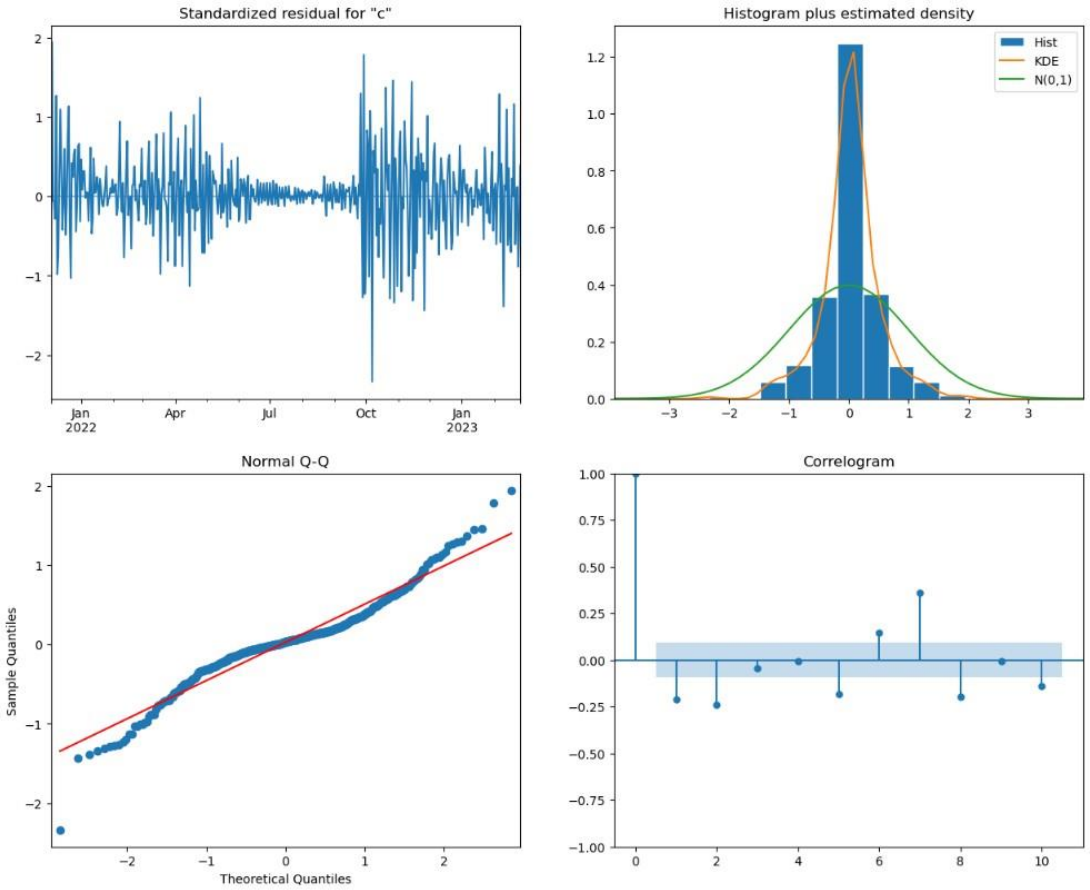


Obrázek 17: Předpovědní výpočty. Zdroj: Vlastní zpracování

```
   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
   11     50     71      1     0     0   1.474D-01   6.065D+00
  F =   6.0650418403198252

STOP: TOTAL NO. of ITERATIONS REACHED LIMIT
                                SARIMAX Results
==========================================================================================
Dep. Variable:                          connUsers   No. Observations:                  477
Model:             SARIMAX(2, 2, 3)x(2, 2, 3, 4)   Log Likelihood               -2893.025
Date:                           Tue, 14 Mar 2023   AIC                           5808.050
Time:                                   14:35:59   BIC                           5853.276
Sample:                             11-07-2021   HQIC                          5825.874
                                  - 02-26-2023
Covariance Type:                            opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.7427      0.083     -8.996      0.000      -0.905      -0.581
ar.L2         -0.8572      0.068    -12.557      0.000      -0.991      -0.723
ma.L1         -0.0599      0.684     -0.088      0.930      -1.400       1.280
ma.L2          0.0628      0.644      0.097      0.922      -1.200       1.325
ma.L3         -0.9943      0.688     -1.446      0.148      -2.343       0.354
ar.S.L4        0.1530      0.401      0.382      0.703      -0.633       0.939
ar.S.L8        0.4005      0.221      1.814      0.070      -0.032       0.833
ma.S.L4       -2.1279      1.132     -1.879      0.060      -4.347       0.091
ma.S.L8        1.4568      1.399      1.041      0.298      -1.285       4.199
ma.S.L12      -0.3283      0.499     -0.657      0.511      -1.307       0.651
sigma2      4.322e+04   4.33e+04      0.998      0.318   -4.17e+04    1.28e+05
==========================================================================================
Ljung-Box (L1) (Q):                  19.88   Jarque-Bera (JB):               156.21
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               1.56   Skew:                            -0.14
Prob(H) (two-sided):                  0.01   Kurtosis:                         5.87
==========================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

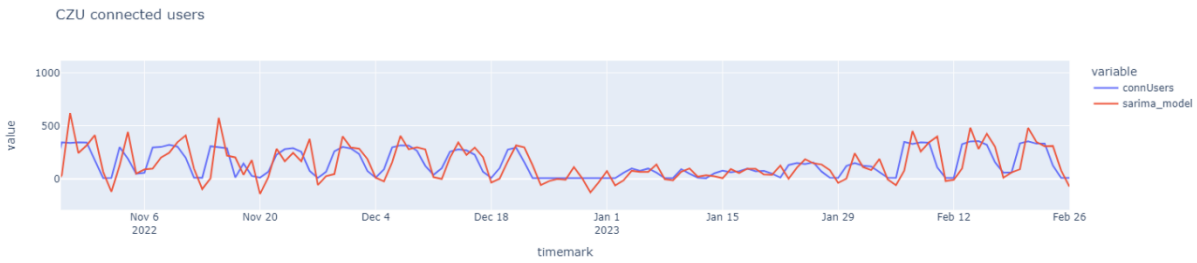
Obrázek 18: Přehled našeho modelu SARIMA. Zdroj: Vlastní zpracování

Můžeme také vizualizovat charakteristiky naší předpovědi:



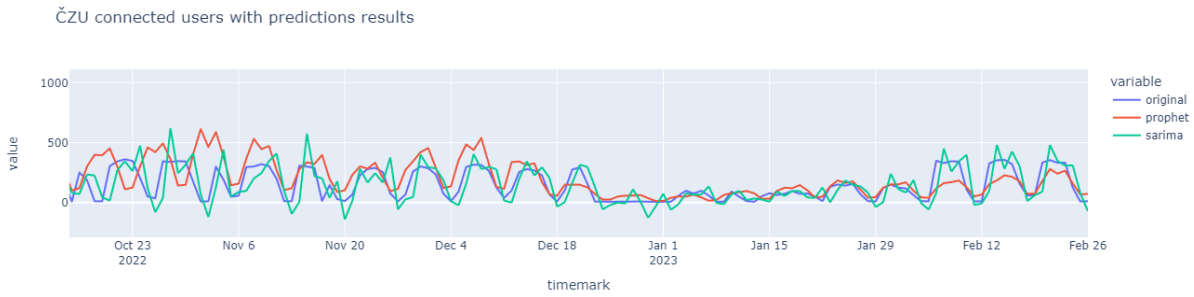
Obrázek 19: Charakteristiky časové řady SARIMA. Zdroj: Vlastní zpracování

Po předpovědi zobrazíme nová data v grafu spolu se skutečnými daty:



Obrázek 20: Predikční graf SARIMA. Zdroj: Vlastní zpracování

5. Výsledky a diskuze



Obrázek 21: Porovnání všech prognóz se skutečností. Zdroj: Vlastní zpracování

Je třeba poznamenat, že sestavení modelu SARIMA je mnohem nákladnější než model Prophet: je třeba prozkoumat původní řadu, uvést ji do stacionární řady, vybrat počáteční aproximace a strávit spoustu času výběrem hyperparametrů pro algoritmus.

MAPE (mean absolute percentage error) je průměrná absolutní chyba naší předpovědi.

MAPE se často používá k hodnocení kvality, protože se jedná o relativní hodnotu a lze ji použít k porovnání kvality i na různých souborech dat.

Kromě toho je někdy užitečné podívat se na MAE (mean absolute error), abychom zjistili, jak moc se model mýlí v absolutním vyjádření.

V tomto případě však nebyla snaha marná a předpověď SARIMA se ukázala jako přesnější: MAPE=15,57 %, MAE=68,93. Nejlepší model s parametry: D=2, d=2, Q=3, q=2, P=2, p=3.

Prophet:

MAPE 19.26082514095201
MAE 96.02685983189524

SARIMA:

MAPE 15.578343269213745
MAE 68.9300912845355

6. Závěr

Naším cílem bylo vytvořit prediktivní model, což se nám podařilo. Podařilo se nám také shromáždit reprezentativní data pro naši analýzu je správně zpracovat a následně použít v prognóze.
Nejlepší model pro naši prognózu byl určen ze dvou vybraných. Podařilo se nám také vytvořit pipelinu pro sběr, zpracování a analýzu našich dat.

Na základě této práce lze vyvodit následující závěry:

- Data a předpovědi lze použít k:
    1. Nalezení nejlepší cesty v rámci univerzity
    2. Optimální rozmístění stánků s občerstvením
    3. Pořádání propagační akce pro univerzitu (na místech, kde je nejvíce lidí)

- Naše pipelina, kterou nyní máme, má obrovský potenciál pro automatizaci a zlepšení.
- Existuje prostor pro další výzkum

7. Seznam použitých zdrojů

1. **WHITE, Tom.** *Hadoop: the definitive guide. 3rd ed.* s.l. : Sebastopol: O'Reilly., 2012. ISBN 978-1-449-31152-0.

2. ***Aruba High Density Wireless networks for Auditoriums.*** **s.l. : VRD. Aruba, 2010.**

3. ***WiFi Networks in Metropolises: From Access Point and User Perspectives.*** **Lei Zhang, Liting Zhao, Zhi Wang, and Jiangchuan Liu. s.l. : IEEE Communications Magazine, 2017.**

4. ***An overview of Technical aspect for WiFi.*** **Kaushik, Shailandra. s.l. : International Journal of Electronics and Computer Science Engineering. ISSN- 2277-1956/V1N1-28-34.**

5. eduroam. [Online] https://eduroam.org/.

6. Prophet - Forecasting at scale. ***facebook.*** **[Online] 2023. https://facebook.github.io/prophet/.**

7. HARRINGTON, Peter. ***Machine learnig in action. Shelter Island: Manning.*** **2012. ISBN 9781617290183.**

8. HASTIE, Trevor J., Robert TIBSHIRANI a J. H. FRIEDMAN. ***The elements of statistical learning: data.*** **2009. ISBN 978-0-387-84857-0.**

9. HENDL, J. ***Přehled statistických metod : analýza a metaanalýza dat.*** **s.l. : Praha: Portál, 2015. ISBN 978-80-262-0981-2.**

10. ***Forecasting at Scale.*** **Letham, Sean J. Taylor and Benjamin. s.l. : THE AMERICAN STATISTICIAN, 2017.**

11. ***Usage Patterns in an Urban WiFi Network.*** **Mikhail Afanasyev, Tsuwei Chen, Geoffrey M. Voelker, Member, IEEE, and Alex C. Snoeren, Member, IEEE. 18-5, s.l. : IEEE/ACM TRANSACTIONS ON NETWORKING, 2010.**

12. SCHUTT, Rachel a Cathy O'NEIL. ***Doing Data Science: Straight Talk from the Frontline.*** **Newton, Massachusets, USA: : O'Reilly Media, 2013. ISBN 978-1449358655..**