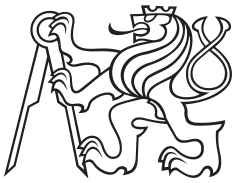
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Podpůrné soubory pro zpracování a analýzu dat

Jakub Klas

Školitel: Ing. Bešťák Robert Ph.D.
Září 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Klas** Jméno: **Jakub** Osobní číslo: **487233**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Podpůrné soubory pro zpracování a analýzu dat

Název bakalářské práce anglicky:

Packages for data processing and analysis

Pokyny pro vypracování:

Cílem práce je vytvořit podpůrné soubory (packages) pro zpracovávání a analýzu velkých dat. Soubory budou tvořeny jednak ze stávajících nebo nově vytvořených funkcí a skriptů. Vytvořené soubory by měly co nejlépe odrážet úkoly, které budou plnit a tím významně zjednodušit práci uživateli při zpracování dat.

Seznam doporučené literatury:

[1] B. Marr. Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance. Wiley. 2015.
[2] Martin Sauter. From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband. Wiley. 2017.
[3] B. Baesens. Analytics in a Big Data World: The Essential Guide to Data Science and its Applications, Wiley. 2014.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Robert Bešťák, Ph.D. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **28.06.2021** Termín odevzdání bakalářské práce: **14.09.2022**

Platnost zadání bakalářské práce: **19.02.2023**

Ing. Robert Bešťák, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji Ing. Robertu Bešťákovi Ph.D. za pomoc při vedení bakalářské práce. Mé poděkováni patří též Etienovi Delort, který mě blízce seznámil s tématikou časových řad a přiblížil mi jejich analýzu. Nakonec bych chtěl poděkovat své rodině a kamarádům za podporu při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 9. září, 2022

Podpis: .....

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, September 9, 2022

Abstrakt

Big data, neboli velká data, je pojem, který označuje velké množství dat, které se nedá zachytit v reálném čase. Jejich využití a analýza je pro firmy velmi důležité z důvodu udržení konkurenceschopnosti na trhu. Tato práce přibližuje jak s velkými daty pracovat a jejich analýzou. Cílem práce je vytvořit prediktivní model, který dokáže předpovídat hodnoty časové řady. Analýza je provedena na datech naměřených na webové stránce. Práce se skládá z pěti kapitol včetně úvodu a závěru. První kapitola je zaměřena na charakteristiku naměřených dat a nástrojům na analýzu. Práce pak pokračuje popisem možností predikce dat a teorii časových řad. Poslední kapitola před závěrem je věnovaná implementaci modelu časové řady.

Klíčová slova: Velká data, Časové řady, Trend, Datframe, predikce, analýza

Školitel: Ing. Bešťák Robert Ph.D.

Abstract

Big data is a term that refers to a large amount of data that cannot be captured in real time. Their use and their analysis is very important for companies in order to maintain their competitiveness on the market. This thesis describes how to work with big data and how to analyze it. The aim of the thesis is to create a model that can predict time series values. The analysis is performed using data measured on a website. It consists of five chapters, including an introduction and a conclusion. The first chapter is focused on the characteristics of measured data and tools of analysis. Then it continues with a description of data prediction possibilities and time series theory. The last chapter before the conclusion is devoted to the implementation of the time series model.

Keywords: Big data, Time series, Trend, Dataframe, prediction, analysis

Title translation: Packages for data processing and analysis

Obsah

Obrázky

Tabulky

Seznam zkratek

ACF	Auto Correlation Function
ADF	Augmented Dickey-Fuller
AIC	Akaike Information Criterion
AR	Auto Regressive
ARMA	Auto Regressive Moving Average
BIC	Bayesian Information Criterion
HQIC	Hannan–Quinn Information Criterion
LLR	Log Likelyhood Ratio
MA	Moving Average
MB	Mega Byte
Nan	Not a Number
PACF	Partial AutoCorrelation Function
SARIMA	Seasonal Auto Regressive Integrated Moving Average

Kapitola 1

Úvod

Velká data je pojem, který se nejčastěji vyskytuje v oblasti informačních technologií a označuje velké množství dat, které se nedá zachytit v reálném čase. Pro představu průměrný člověk v roce 2021 vytvořil 1,7 MB dat za sekundu. Jako velká data si člověk může představit víceméně cokoliv, co se dnes pohybuje volným internetem, mohou to být data získaná ze sociálních sítí, klientských databází nebo e-shopů. Data tím pádem nemusí mít stejnou strukturu, a proto je potřeba je analyzovat, abychom s nimi mohli dále pracovat.X

Postupy, jak větší množství informací ukládat a používat k analýze, se aplikují už delší dobu. Avšak koncept velkých dat, jak je známe dnes, se začal používat až na začátku 21. století, kdy datový analytik Doug Laney [1] vytvořil dnešní definici velkých dat pomocí základních 3 velkých písmen V Objem (Volume), Rychlost (Velocity) , Různorodost (Variety), které značí a popisují jejich vlastnosti.

Používání velkých dat je v dnešní době ve většině průmyslových odvětvích velmi běžné. Tento koncept je pro společnost velmi důležitý z důvodu udržení konkurenceschopnosti na trhu. Moje práce se bude věnovat právě analýze velkých dat, konkrétně datům sesbíraných z webových stránek. Tyto informace v sobě obsahují denní počet připojených uživatelů k této stránce v průběhu necelých tří let. To je celkem 110 681 760 připojených uživatelů. Toto číslo není ale kompletní, protože z tabulky s daty byla odstraněná chybně naměřená data, neboli anomálie. Takže počet celkově připojených je určitě vyšší, vzhledem k tomu, že za anomálie bylo označeno 82 dní. Při průměrném počtu 120 000, můžeme předpokládat, že celkový počet připojených bude v měřeném období okolo 120 000 000.

Cílem práce je analyzovat sesbíraná data využitím prediktivních teorií, např. teorie časových řad. Pomocí této teorie se snažím data vyčistit a následně vytvořit prediktivní model, který odhadne chybějící hodnoty mezi daty. Tento model budu porovnávat s reálně naměřenými hodnotami, aby odhadnutá data odpovídala co nejlépe realitě.

V teoretické části bakalářské práce popisuji sesbíraná data a mnou použité technologie pro jejich analýzu. V následující části dokumentu vám představím teorie časových řad společně s jinými modely predikce hodnot a testy

použitými pro porovnávání modelů. V předposlední kapitole se zabývám implementací modelů a zároveň se věnuji porovnávání prediktivních modelů a jejich výstupům.

Kapitola 2

Data

V této kapitole rozeberu data, která jsem zkoumal. Budu zde popisovat, jaká konkretní data jsem na analýzu použil, jaké je časové období mého zkoumání a jaké nedokonalosti moje data mají. Následně vám popíši prostředí programů Anaconda a Jupyterlab, ve kterých jsem analýzu provedl. V poslední části kapitoly představím použité knihovny potřebné pro práci s daty a jejich analýzu.

2.1 Zkoumaná data

Naměřená data, která jsem pro svoji práci použil jsem obdržel od vedoucího práce ve formátu .csv a představují dvourozměrnou tabulku. Tabulka 2.1 s daty obsahuje hodnotu připojených uživatelů na webovou stránku v jednotlivých dnech, a to od 1.7.2019 do 31.3.2022. První sloupec představuje dny sesbíraných dat a druhý sloupec představuje naměřenou hodnotu připojených uživatelů na webovou stránku. Tabulka s hodnotami má celkem 923 záznamů, z celkového rozpětí 1005 dní. To znamená, že bylo z dat odstraněno 82 záznamů, které byly označeny jako anomálie. Pro zachování integrity dat, nesmím šířit dále přesnější informace o webové stránce nebo o datech samotných.

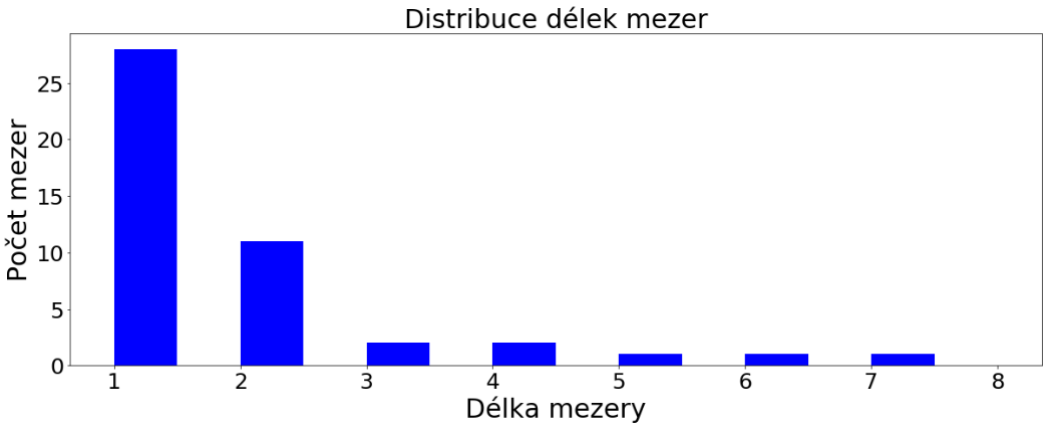
V pozorovaném období proběhla pandemie Covidu-19 [2], jejíž vlny silně ovlivnily počty připojených uživatelů a tudíž data naměřená při nouzových stavech nemusejí odpovídat datům v době bez celosvětové pandemie. Například období od 29.02.2020 do 25.05.2020 odpovídá období vypuknutí pandemie Covidu-19 v České republice. Pro mnou analyzovaná data to znamená, že naměřená data v tomto období jsou relativně vyšší než v období předchozím. Nejenom pandemie ovlivňuje počet připojených uživatelů, při takto krátkém období musíme při analýze a následné predikci těchto dat, brát v potaz i jiné proměnné. Těmito proměnnými mohou být například jednotlivé dny v týdnu, víkendy, státní svátky aj.

Jako **anomálii** můžeme chápat hodnotu dat, která nevyhovuje okolním hodnotám. Takové hodnoty mohou vzniknout mnoha způsoby a není možné na sto procent určit, jak k anomálii došlo a kde nastala chyba, že hodnota nabyla tak nepřiměřeně velkého nebo malého čísla. Z tabulky2.1 si například

Tabulka 2.1: Ukázka pozorovaných dat ze dnů od 1.7.2019 do 19.7.2019

Datum	Počet připojených uživatelů
01.07.2019	82836
02.07.2019	78176
03.07.2019	77050
04.07.2019	73964
05.07.2019	103506
06.07.2019	108434
07.07.2019	115626
08.07.2019	79432
09.07.2019	77950
10.07.2019	74000
11.07.2019	75698
12.07.2019	73576
13.07.2019	105194
14.07.2019	113082
15.07.2019	74898
16.07.2019	34834
18.07.2019	50754
19.07.2019	60198

můžeme povšimnout chybějícího dne 17.7.2019, který byl z dat odstraněn. Tyto hodnoty byly odstraněny pomocí detekčního programu, který postupně porovnává hodnoty s předchozími naměřenými hodnotami a podle toho určí, jestli je hodnota anomálie. Z histogramu 2.1 můžeme vidět rozložení mezer mezi naměřenými hodnotami. Touto mezerou je myšlený souvislý počet chybějících hodnot v datech. Na osách grafu nalezneme délku mezery ve dnech a počet mezer dané velikosti.



Obrázek 2.1: Histogram rozložení mezer

Velikost těchto mezer je pro predikci velice důležitá, protože predikování hodnoty daného dne je v naší časové řadě silně ovlivněno hodnotou předchozího

dne. Tím pádem predikce na velké mezeře může být výrazně nepřesnější, než predikce na mezeře malé.

2.2 Nástroje pro analýzu

V této sekci vám podrobněji popíši prostředí, které jsem pro svoji analýzu použil. Tím byl programovací jazyk Python, který jsem spouštěl pomocí webového prostředí Jupyterlab, hostovaného na localhostu pomocí programu Anaconda. Pomocí programu Anaconda jsem si vytvořil prostředí, ve kterém jsou uloženy potřebné knihovny pro funkčnost mého programu. Toto prostředí je uložené na githubu, takže je možné ho znovu použít.

2.2.1 Anaconda

Anaconda je vývojové prostředí pro jazyky Python a R, používané na datové vědy. Hlavní výhodou tohoto prostředí je management knihoven a packege, pomocí prgoramu Conda. V tomto programu si uživatel může vytvořit vlastní prostředí, do kterého si nastaví vlastní potřebné knihovny a programy, které bude na projektu využívat.

Anaconda Navigator[3] je desktopové grafické uživatelské rozhraní zahrnuté v distribuci Anaconda, které umožňuje uživatelům spouštět aplikace a spravovat balíčky Conda, prostředí a kanály bez použití příkazů v příkazovém řádku. Navigator může vyhledávat balíčky na Anaconda Cloud nebo v místním úložišti Anaconda, ty následně instalovat do prostředí, spouštět a aktualizovat je.

Conda [4]je open source, multiplatformní, jazykový agnostický správce balíčků a systém správy prostředí, který instaluje, spouští a aktualizuje balíčky a jejich závislosti. Byl vytvořen pro program Python, ale může dávat do balíčků a distribuovat software pro jakýkoli jazyk (např. R), včetně vícejazyčných projektů.

2.2.2 Jupyterlab

Jupyterlab je webové vývojářské prostředí od firmy Project Jupyter. JupyterLab vám umožňuje pracovat s dokumenty a aktivitami, jako jsou notebooky, textové editory, terminály a vlastní komponenty, z nichž nejpoužívanější je Jupyter Notebook.

Jupyter Notebook [5] vznikl z neméně známého a používaného projektu IPython Notebook(s). Využívá interaktivní prostředí, které se zobrazuje přímo ve webovém prohlížeči a obsahuje klasickou smyčku REPL (Read–Eval–Print–Loop), což znamená, že se jednotlivé výrazy zapsané uživatelem mohou ihned vyhodnocovat s prakticky okamžitou zpětnou vazbou. Celé grafické uživatelské rozhraní Jupyter Notebooku napodobuje diář (notebook), do kterého se zapisují jak poznámky, tak i případný programový kód a jeho výsledek, takže se tento systém může hodit i pro tvorbu interaktivních prezentací.

2.2.3 Python

Python[6] je interpretovaný, objektově orientovaný, vysokoúrovňový programovací jazyk s dynamickou sémantikou. Díky vestavěným datovým strukturám na vysoké úrovni v kombinaci s dynamickým psaním a dynamickým vázáním je velmi atraktivní pro rychlý vývoj aplikací a také pro použití jako skriptovací nebo spojovací jazyk pro sloučení existujících komponent dohromady. Jednoduchá snadno naučitelná syntaxe Pythonu klade důraz na čitelnost, a proto snižuje náklady na údržbu programu. Python podporuje moduly a balíčky, což umožňuje modularitu programu a opětovné použití kódu. Interpret Pythonu a rozsáhlá standardní knihovna jsou k dispozici ve zdrojové nebo binární podobě zdarma pro všechny hlavní platformy a lze je volně šířit.

2.2.4 Balíčky pro datovou analýzu

V této sekci vám popíši základní knihovny pro datovou analytiku v programovacím jazyce Python. Tyto balíčky jsou jedny z nejpoužívanějších nástrojů pro analýzu dat v oboru datových věd. Proto jsem se je rozhodl použít v této práci. Budu zde charakterizovat knihovny Numpy, Pandas, Matplotlib, Statsmodels a Scipy.

NumPy[7] je balíček pro zpracování polí v Pythonu a poskytuje vysoce výkonný vícerozměrný objekt pole (numpy array) a nástroje pro práci s těmito objekty. Je to základní balíček pro vědecké výpočty s Pythonem.

NumPy Array je tabulka prvků (obvykle čísel), všechny stejného typu, indexované n-ticí kladných celých čísel. V Numpy se počet rozměrů pole nazývá hodnost pole. N-tice celých čísel udávajících velikost pole podél každé dimenze se nazývá tvar pole.

Indexování lze provést v NumPy pomocí pole jako indexu. Numpy pole lze indexovat s jinými poli nebo jakoukoli jinou sekvencí s výjimkou n-tic. Poslední prvek je indexován jako -1 ,předposlední -2 a tak dále.

Python Pandas[8] se používá pro relační nebo označená data a poskytuje různé datové struktury pro manipulaci s takovými daty a časovými řadami. Tato knihovna je postavena na knihovně NumPy. Pandas obecně poskytuje dvě datové struktury pro manipulaci s daty, těmi jsou:

Pandas Series je jednorozměrné pole schopné pojmout data libovolného typu (celé číslo, řetězec, float, objekty v pythonu atd.). Popisky os se souhrnně nazývají indexy. Pandas Series není nic jiného než sloupec v listu aplikace Excel. Štítky nemusí být jedinečné, ale musí být hašovatelného typu. Objekt podporuje celočíselné indexování i indexování založené na štítcích a poskytuje řadu metod pro provádění operací zahrnujících indexy.

Pandas DataFrame je dvourozměrná, velikostně proměnlivá, potenciálně heterogenní tabulková datová struktura s označenými osami (řádky a sloupce). Datový rámec je dvourozměrná datová struktura, tj. data jsou zarovnána tabulkovým způsobem do řádků a sloupců. Pandas DataFrame se skládá ze tří hlavních komponent: data, řádky a sloupce. Proto se velmi hodí pro analýzu časových řad, které se nejčastěji zapisují v podobě tabulky nebo grafu.

Knihovna **Matplotlib**[9] slouží k vytváření statických, animovaných a interaktivních vizualizací v Pythonu. Je postaven na polích NumPy a navržen pro práci s širším zásobníkem SciPy a skládá se z několika grafů, jako je přímka, sloupec, rozptyl, histogram atd.

Pyplot je modul Matplotlib, který se podobá rozhraní MATLABu. Pyplot poskytuje funkce, které interagují s obrazci, tj. vytváří grafy, popisuje graf, vytváří vykreslovací plochu obrazce.

Histogram se používá k reprezentaci dat ve formě skupin. Je to typ sloupcového grafu, kde osa X představuje možnosti rozsahů, zatímco osa Y poskytuje informace o frekvenci. Chcete-li vytvořit histogram, prvním krokem je vytvořit možnosti rozsahů, poté rozložit celý rozsah hodnot do řady intervalů a spočítat hodnoty, které spadají do každého z intervalů. Příklad intervalu si můžeme všimnout na obrázku 2.1.

Statsmodels [10] je postaven na numerických knihovnách NumPy a SciPy, integruje se s Pandas pro práci s daty a používá Patsy pro rozhraní vzorců podobné jazyku R. Grafické funkce jsou založeny na knihovně Matplotlib. Statsmodels poskytuje statistický backend pro další knihovny Pythonu. jeho hlavní využití je vytváření statistických modelů, vytváření statistických testů a statistické prozkoumávání dat.

Knihovna **Scipy** [11] je nadstavbou knihovny Numpy, pro práci s daty nabízí dodatečné nástroje na práci s poli. Nabízí také možnost pracovat se složitějšími datovými strukturami. Základní datová struktura používaná knihovnou SciPy je vícerozměrné pole poskytované modulem NumPy. NumPy poskytuje některé funkce pro lineární algebru, Fourierovy transformace a generování náhodných čísel, ale ne s obecností ekvivalentních funkcí jako ve SciPy.

Kapitola 3

Metody predikování dat

V této kapitole popíši pojmy, které tvoří základ pro teoretickou část práce. V rámci kapitoly jsou přiložené vzorce pro lepší ilustraci matematických pojmů. V první sekci popíši metody predikce pomocí interpolace. Tato metoda je velmi důležitým krokem při vyhlazování dat, a je stavebním kamenem pro předpovídání hodnot pomocí strojového učení. Zároveň popíši 2 druhy interpolace, lineární a polynomiální, které jsem při práci s daty používal.

V další části se budu věnovat časovým řadám a popíši, co si jako časovou řadu můžeme představit a jak se časové řady dělí. Poté vysvětluji různé metody pro určování trendu časové řady. Eliminace trendu je důležitá pro přesnost předpovězených hodnot časové řady, nejlépe chceme z řady odstranit všechny složky, které dělají řadu nestacionární. Stacionárnost řady je významná u prediktivních modelů AR, MA a ARMA. Metoda časových řad mi přišla jako ideální volba pro predikci mnou získaných dat.

Dále jsou vyjmenované prediktivní modely, které spadají pod metodiku Box-Jenkins a funkce, které jsou podstatné pro porovnání modelů. Funkce ACF a PACF potřebujeme, abychom zjistili, který model nejlépe určuje počet zpoždění při AR, MA, ARMA modelech. V poslední kapitole této sekce popisuji testy, které jsem použil na porovnávání jednotlivých modelů, tím byl LLR test a model na zjištění stacionarity AR modelu a tím byl Dickey-Fuller test. Teoretický rámec definovaný v této kapitole mi umožní k aplikaci těchto metod v praktické části.

3.1 Interpolace

V numerické matematice pojem interpolace znamená nalezení přibližné hodnoty funkce na daném intervalu. Interpolace je jednou z metod aproximace funkce. Úlohou aproximace je tedy nalezení jednodušší a matematicky přesně definované spojité aproximační funkce F_x v intervalu (a, b), která by co nejlépe přiléhala k empirickým bodům x_0,x_1,...,x_n.

Interpolace jako taková se nehodí pro předpovídání hodnot do budoucnosti, protože pokud jistě neznáme krajní body intervalu, tak nemusí být hodnoty, ani přibližně přesné. Pro svojí práci jsem si vybral nejjednodušší z nich, kterou je lineární interpolace.

3.1.1 Lineární interpolace

Lineární interpolace[12] je taková funkce, která si na začátku zvolí krajní dva body intervalu, na kterém chceme data předpovídat a proloží jimi přímku, neboli interval aproximuje polynomem. Při lineární interpolaci aproximuje polynomem prvního stupně se vzorcem 3.1,

$$L_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \tag{3.1}$$

kde $[x_0, y_0]$ a $[x_1, y_1]$ jsou souřadnice známých bodů funkce. A $f(x_0), f(x_1)$ jsou funkční hodnoty v těchto bodech.

3.1.2 Polynomiální interpolace

Polynomiální interpolace je zobecněním interpolace lineární. Zde neaproximujeme interval funkce polynomem prvního stupně, ale aproximujeme polynomem n-tého stupně. V případě, že máme n bodů funkce které známe, tak existuje pouze jeden polynom n-tého stupně, neboli předpis funkce. Vzorec kvadratické interpolace neboli aproximace polynomem 2. stupně vypadá viz. 3.2. Hodnoty proměnných vycházejí ze vzorce pro lineární interpolaci.

$$\begin{aligned} L_2(x) = {} & \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \\ & + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \end{aligned} \tag{3.2}$$

3.2 Časové řady

Časová řada je posloupnost datových bodů, které se vyskytují za sebou v určitém časovém období. Jako časovou řadu můžeme označit data, která jsou uspořádána chronologicky v čase. Mohou jí tedy být mnou obdržená data, nebo například vývoj teploty každý den v nějakém městě. Pro časové řady je velmi důležité, aby její data byla zatížena nejistotou. To znamená, že nejsou deterministická, protože deterministická data se dají jednoznačně popsat nějakým konkrétním matematickým vzorem. Pokud tedy použijeme teorii náhodných procesů, můžeme o časové řadě říci, že představuje konkrétní realizaci náhodného stochaistického procesu[13].

3.2.1 Dekompozice časové řady

Dekompozice časové[14] řady je jeden z přístupů analýzy časové řady. Dekompozice rozděluje časovou řadu na 4 části: Trend, Sezónní složku, Cyklickou složku a Náhodnou reziduální složku.

Trend

Jako trend můžeme chápat dlouhodobé změny v průměrném chování řady, například dlouhodobý růst nebo dlouhodobý pokles, a nebo dlouhou konstantní úroveň.

Sezónní složka

Sezonní složka popisuje periodické změny v časové řadě. Tyto změny zpravidla souvisejí se střídáním ročních období, nebo s jinou pravidelnou činností opakující se každý rok.

Cyklická složka

Cyklická složka popisuje fluktuace kolem trendu. Zde se pravidelně střídají fáze růstu s fázemi poklesu. Délka jednoho cyklu se může průběhem času měnit. Vznik této složky je přitom velmi těžký definovat, jeden cyklus může trvat i několik let.

Náhodná reziduální složka

Tato složka představuje náhodné fluktuace. Jejich charakter je zpravidla nesystematický. Zahrnuje totiž chyby v měření nebo ve statistickém zpracování dat. Můžeme předpokládat, že tato složka má charakter bílého šumu, což znamená, že je tvořena hodnotami nezávislých náhodných veličin s nulovou střední hodnotou a konstantním rozptylem.

Dekompozice se dá provést dvěma základními způsoby (a modifikacemi těchto způsobů). První způsobem je aditivní tvar. Zde uvažujeme, že časová řada je součtem jednotlivých složek viz 3.3, kde T_t je absolutní hodnota trendu časové řady v čase t, S_t je absolutní hodnota sezónní složky v čase t, C_t je absolutní hodnota cyklické složky v čase t a ϵ_t je absolutní hodnota reziduální složky časové řady v čase t.

$$y_t = T_t + S_t + C_t + \varepsilon_t, \quad t = 1, 2, ..., n. \tag{3.3}$$

Druhým způsobem je multiplikativní tvar, kde je pouze trendová hodnota ponechána ve své absolutní hodnotě a ostatní složky jsou v hodnotách relativních.

$$y_t = T_t S_t C_t \varepsilon_t, \quad t = 1, 2, ..., n. \tag{3.4}$$

3.2.2 Určování trendu

Trendová složka hraje v časových řadách velmi významnou rol. V následující sekci se budu věnovat způsobům, jak trendovou složku eliminovat. Existují dva základní přístupy k eliminaci trendu:

- Klasické přístupy (matematické, analytické přístupy)
- Adaptivní přístupy

Klasické přístupy

Klasické přístupy zahrnují snahu o popsání trendu na základě některých matematických funkcí. Pro odhad parametrů funkcí se zpravidla používá metoda nejmenších čtverců. Ta se používá hlavně v případě, kdy je trendová složka v lineárních parametrech. Jde tedy o lineární regresní model.

Pomocí metody nejmenších čtverců u konstantní funkce, tj. minimalizací funkce

$$S = \sum_{t=1}^{n} (Y_t - \beta_0)^2 \tag{3.5}$$

dostaneme pro odhad bodu b_0 parametru β_0 vztah

$$b_0 = \frac{1}{n} \sum_{t=1}^{n} y_t = \overline{y} \tag{3.6}$$

Trend konstantní funkce potom vypadá takto:

$$T_r = \beta_0, \quad t = 1, 2, ..., n. \tag{3.7}$$

V ekonomii nejpoužívanější funkce:

- **Lineární trend**

  Výsledek lineárního trendu vychází z konstantního trendu,

  $$T_r = \beta_0 + \beta_1 t, \quad t = 1, 2, ..., n. \tag{3.8}$$

  parametry získáme použitím metody nejmenších čtverců a následně vyřešením soustavy rovnic.

- **Parabolický trend**

  Parabolický neboli kvadratický trend vychází ze dvou předchozích případů. Zde potřebujeme vyřešit soustavu 3 rovnic abychom se dostali ke vztahu parametrů β a bodů b.

  $$T_r = \beta_0 + \beta_1 t + \beta_1 t^2, \quad t = 1, 2, ..., n. \tag{3.9}$$

- **Exponenciální trend**

  Exponenciální trend není v parametrech lineární, proto je potřeba převést pomocí logaritmické transformace na lineární trend. Na zlogaritmované hodnoty provedeme metodu nejmenších čtverců a získáme logaritmované odhadnuté parametry. Po jejich odlogaritmování získáme odhady parametrů. Zde je navíc parametr α.

  $$Tr_t = \alpha\beta^t, \; \alpha > 0, \; \beta > 0, \quad t = 1, 2, ..., n. \tag{3.10}$$

Adaptivní přístupy

Adaptivní přístupy se při eliminaci trendu od těch klasických liší přístupem k parametrům v čase. Zatímco u klasických přístupů se trend popisoval v celém časovém období pomocí jedné matematické funkce, což znamená, že parametr je konstantní na celém časovém rozmezí. U adaptivních přístupů se počítá s nestabilitou analytického tvaru a nekonstantními parametry v čase. Proměnlivost parametrů v čase je docílena předpokladem, že na krátkých časových úsecích časové řady je možné vyrovnání pomocí matematické funkce. Tyto krátké úseky se poté můžou lišit v hodnotách parametrů. Metodou adaptivních přístupů je například metoda klouzavých průměrů.

Metoda klouzavých průměrů spadá do adaptivního přístupu eliminace trendů. To znamená, že pracuje s takovými časovými řadami, jejichž trend může podléhat časovým změnám. Tato metoda tedy neaproximuje celou časovou řadu, ale pouze její úseky pomocí polynomu nízkého stupně. Metoda klouzavých průměrů je založena na vyrovnávání krátkých úseků časové řady polynomickými funkcemi. Má dva parametry: délku klouzavých průměrů a řád klouzavých průměrů. Délka klouzavých průměrů udává skutečnou délku vyrovnávaných úseků časové řady. Předpokládá se, že je to liché číslo. Řád klouzavých průměrů (r) reprezentuje stupeň vyrovnávacího polynomu.

3.2.3 Boxova-Jenkinsova metodologie

Pro popis trendu časové řady můžeme též použít jiné metody, například metodologii Box-Jenkins, která považuje za základní prvek modelování náhodnou složku. Abychom tedy mohly metodologii použít, budeme potřebovat, aby časová řada byla stacionární.

Stacionarita

V Boxově – Jenkinsově metodologii lze modelovat pouze stacionární časové řady, resp. takové řady, které mohou být převedeny na stacionární. V teorii se rozlišuje dvojí stacionarita, striktní a slabá. Striktní stacionarita předpokládá, že chování příslušného náhodného procesu, tj. jeho rozdělení, je invariantní vůči časovým posunům. Naproti tomu slabá stacionarita je méně omezující; požaduje se, aby příslušný náhodný proces měl konstantní střední hodnotu, konstantní rozptyl a pro kovariance platilo

$$cov(Y_t, Y_s) = cov(Y_{\mathrm{t+h}}, Y_{\mathrm{s+h}}) \tag{3.11}$$

a to pro libovolné h. Uvedený vztah představuje požadavek, aby závislost mezi dvěma libovolnými pozorováními závisela jen na jejich časové vzdálenosti a nikoli na jejich časovém umístění v řadě[13].

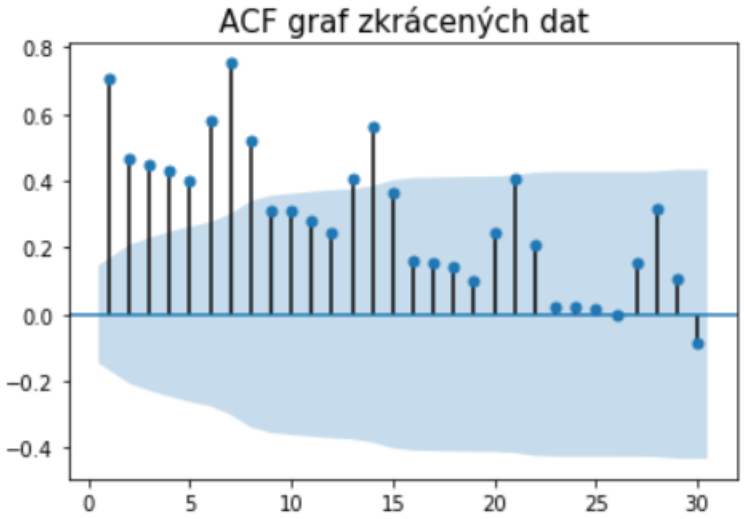
Autokorelační funkce a Parciální atuokorelační funkce

Autokorelační funkce (označuje se také jako ACF) nám dává závislost hodnot v čase. Porovná závislost veličin y_{t} a $y_{\mathrm{t\text{-}k}}$ a lze ji popsat jako podobnost mezi

pozorováním v závislosti na časové prodlevě mezi nimi. Vzorec na výpočet autokorelační funkce se zpožděním k je podíl autokovariační funkce a rozptylu:

$$\rho_k = \frac{cov(X_t, X_{\text{t+k}})}{D(X)} = \frac{\gamma_k}{\gamma_0} \tag{3.12}$$

O autokorelační funkci můžeme hovořit pouze v případě, že je hodnocena závislost hodnot jednoho náhodného procesu ve dvou různých časových okamžicích. Grafem ACF je korelogram. Ten značí na škále od -1 do 1 jak moc hodnota daného zpoždění ovlivňuje hodnotu současnou. Z obr. 3.1 lze vidět, že současnou hodnotu nejvíce ovlivňuje hodnota předchozí a hodnota týden stará, tedy 7 dní zpátky.



Obrázek 3.1: Graf ACF

Průběh autokorelační funkce je důležitým faktorem při výběru vhodného modelu pro danou časovou řadu. Přitom je nutno určit takovou hodnotu p, za kterou začíná být teoretická autokorelační funkce nulová, nebo zjistit, zda taková hodnota vůbec existuje.

Korelace mezi dvěma náhodnými hodnotami spočítána pomocí ACF, je velmi často ovlivněna působením jiné veličiny nebo více jiných veličin. Parciální autokorelace (značená PACF) nese pouze informaci o korelaci veličin y_{t} a $y_{\text{t-k}}$ bez vlivu veličin mezi sebou. To znamená, že pokud chceme spočítat závislost mezi hodnotami y_{t} a $y_{\text{t-2}}$, nebude tato závislost ovlivněna hodnotou $y_{\text{t-1}}$.

Autoregresní model

Autoregresní model[15] řádu p neboli AR(p) je technika lineárního prediktivního modelování, kdy hodnota časové řady v čase t je předpovězena lineární kombinací minulých hodnot této řady (jejich počet značíme p). Její zápis vypadá takto:

$$y_t = c + \phi_1 y_{\text{t-1}} + \phi_2 y_{\text{t-2}} + ... + \phi_p y_{\text{t-p}} + \epsilon_t \tag{3.13}$$

Kde ϵ_t je bílý šum v čase t nebo reziduální error a ϕ_n je koeficient, který popisuje, jak moc daná předchozí hodnota ovlivňuje výsledek současné hodnoty.

Model klouzavých průměrů

Na rozdíl od autoregresního modelu, model klouzavých průměrů nepoužívá minulé hodnoty k predikci dat, ale namísto nich používá chyby v predikcích předchozích hodnot.

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \tag{3.14}$$

Značí se jako MA(q) (moving average model), kde q je počet chyb, ze které se naše hodnota počítá, ϵ_t je bílý šum v čase t a θ_n je koeficient.

Model ARMA

Pokud zkombinujeme předchozí dva modely, dostaneme ARMA model neboli autoregressive–moving-average model. Značí se ARMA(p,q), kde p a q je počet předchozích hodnot, ze kterých se náš model skládá. ARMA je tedy kombinací dvou polynomů, AR a MA. Při odhadu spektra parametrů ARMA[16] se nejprve odhadnou parametry AR a poté se na základě těchto parametrů AR odhadnou parametry MA. Poté jsou získány spektrální odhady modelu ARMA. Odhad parametrů MA modelu se proto často vypočítává jako proces asociace spektra parametrů ARMA. Tento model se často používá v mechanice, kde slouží k diagnostice a analýze poruch, protože dokáže zpracovat samostatné frekvence sinusového signálu.

3.3 Testování modelu

V následující sekci popíši dvě metody statistického testování časových řad. Prvním je test LLR, který se používá na porovnávání modelů časových řad. Tento test porovnává 2 konkrétní modely s určitou délkou zpoždění, jako jsou například AR(1) a AR(2) a určuje změnu předpovězených hodnot obou modelů. Tím druhým je Dickey-Fuller test, který slouží k vyjádření stacionarity modelu.

3.3.1 LLR test

LLR test[17] statisticky porovnává shodu jednoho modelu s druhým. Odstranění prediktorových proměnných z modelu téměř vždy způsobí, že model bude hůře pasovat (tj. model bude mít nižší logaritmickou pravděpodobnost), ale je nutné otestovat, zda je pozorovaný rozdíl v přizpůsobování modelu statisticky významný. LLR test to dělá porovnáním logaritmické pravděpodobnosti dvou modelů, pokud je tento rozdíl statisticky významný, pak se říká, že méně restriktivní model (ten s více proměnnými) vyhovuje datům výrazně lépe než více restriktivní model. Pokud máme logaritmické pravděpodobnosti z

modelů, lze test LLR poměrně snadno vypočítat. Vzorec pro statistiku testu LLR je:

$$LR = -2ln(\frac{L(m_1)}{L(m_2)}) = 2(loglik(m_2) - loglik(m_1)) \tag{3.15}$$

kde L(m) označuje pravděpodobnost příslušného modelu a loglik(m) značí přirozený logaritmus konečné pravděpodobnosti modelu (tj. logaritmickou pravděpodobnost). Kde m_1 je více restriktivní model a m_2 je méně restriktivní model.

3.3.2 Dickey-Fuller Test

Ve statistice testuje Dickey-Fullerův[18] test nulovou hypotézu, že v autoregresním modelu časové řady je přítomen jednotkový kořen. Alternativní hypotéza se liší v závislosti na použité verzi testu, ale obvykle je stacionární nebo trendová. Vzorec jednoduchého D-F testu je:

$$y_t = c + \beta t + \alpha y_{\text{t-1}} + \Phi \Delta Y_{\text{t-1}} + e_t \tag{3.16}$$

kde $y_{(\text{t-1})}$ je zpoždění časové řady a $\Delta Y_{\text{t-1}}$ je první rozdíl řady v čase $t-1$. Výsledek testu obsahuje hodnotu p, která udává pravděpodobnost měřící důkaz proti nulové hypotéze. Čím nižší pravděpodobnost, tím silnější důkaz proti nulové hypotéze.

Chceme-li určit, zda jsou data stacionární, je třeba porovnat výsledek testu s kritickou hodnotou nebo hodnotu p na zvolené hladině. Vzhledem k tomu že, hodnota p obsahuje více aproximace, doporučuje se pro analýzu použít posouzení nulové hypotézy kritickou hodnotu. Ta záleží na zvolené hladině (většinou 0.01, 0.05 nebo 0.10). Obvykle je ale závěr pro obě hodnoty stejný. Pokud vyjde číslo vyšší než je kritická hodnota, test potvrzuje nulovou hypotézu a tím pádem je řada nestacionární. V opačném případě je řada stacionární.

Kapitola 4

Implementace

Tato kapitola se věnuje popisu praktické části práce. Budu zde popisovat, jak jsem si vytvořil dané prostředí v programu Anaconda, které je dostupné na githubu. Následně zde popíši způsob, jakým jsem vytvářel jednotlivé prediktivní modely. A na závěr vysvětlím jak modely fungovali a jaký model se hodí pro predikci dat.

4.1 Návrh prostředí

Na vytvoření prostředí pro analýzu dat jsem použil program Anaconda Navigator. V tomto programu jsem použil podpůrný program Conda Prompt, který slouží k ovládání prostředí z příkazové řádky. Pomocí Conda Prompt jsem si stáhnul do Anaconda Navigator potřebné balíčky a následným složením těchto balíčků jsem si vytvořil prostředí nazvané cleaning neboli čištění. Toto prostředí jsem během své práce několikrát upravil, kvůli doplňování verzí balíčků. Pomocí tohoto prostředí jsem poté v Anacondě spustil webové vývojářské prostředí Jupyterlab, které následně obsahovalo tyto balíčky. Pro použití mého prostředí doporučuji mít stažený program Anaconda Navigator, protože prostředí je vyexportované ve formátu .yml, které s Anacondou velmi dobře spolupracuje.

4.2 Vytvořené scripty

V následující sekci popíši, co obsahují mnou naprogramované scripty, a jak jsem jednotlivé metody použil pro analyzování a predikci dat. Script se vždy nejprve věnuje grafickému znázornění časové řady, následně dekompozici této řady. Po dekompozici řady probíhá statistické testování a nakonec trénování a vytváření prediktivního modelu.

4.2.1 Příprava dat

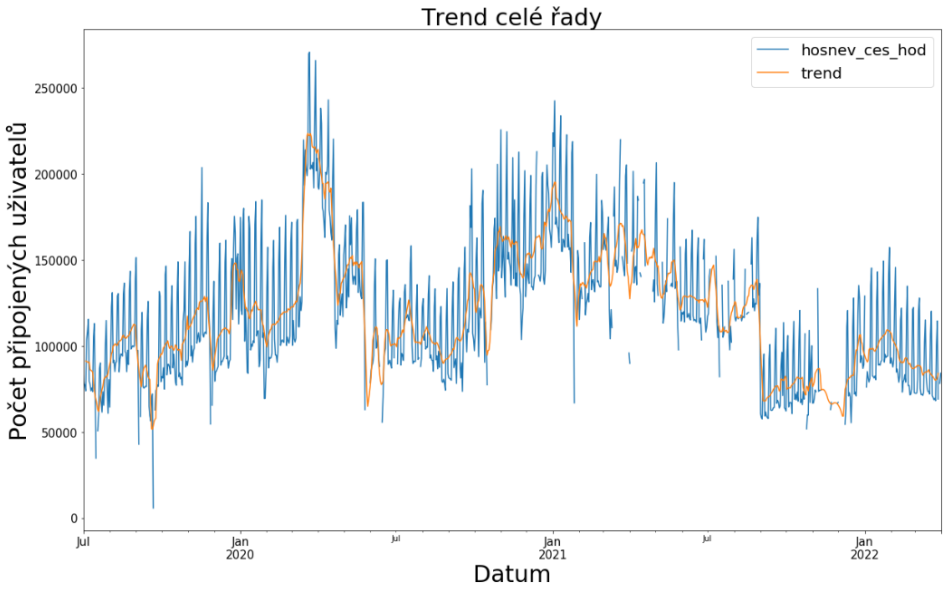
Do programu Jupyterlab jsem si pomocí balíčku Python Pandas, který je popsaný v kapitole číslo 2.2.4, nahrál mnou obdržená data do objektu TSDataframe. Tento objekt je nadstavbou objektu dataframe z balíčku Pandas,

který jsem obohatil o funkce používané k mé analýze. První z těchto funkcí je například metoda, která doplní chybějící řádky v tabulce. Celou tabulku porovná funkce s reálným kalendářem, a pokud funkce zjistí, že nějaké datum chybí, toto datum do tabulky doplní a nastaví jeho hodnotu na not a number, neboli Nan. Následující funkce ve třídě TSDataframe je použitá na vytvoření histogramu mezer v grafu. Těmito mezerami jsou myšlené dny, které nemají hodnotu nebo je jejich hodnota nastavená právě na hodnotu Nan. Funkce vypočítá počet jednotlivých mezer a jejich délku, kde délkou mezery je myšlený spojitý počet dní bez hodnoty. Další mnou vytvořená funkce přidaná do zmiňované třídy vypočítá interpolaci celé tabulky. Tato funkce má za úkol doplnit do tabulky chybějící hodnoty. Funkce iteruje přes celou tabulku, kde každé chybějící hodnotě najde x nejbližších nechybějících hodnot a na nich spočítá hodnotu pomocí interpolace. Číslo x záleží od velikosti polynomu použitého k interpolaci. V mém případě jsem zvolil lineární interpolaci což znamená, že nám stačí polynom prvního stupně, x bude tedy rovno dvěma.

4.2.2 Analýza dat

Po doplnění dat pomocí lineární interpolace jsem se rozhodl data blíže analyzovat. Hned zpočátku je na datech vidět opakující se týdenní vzor. Hodnoty v týdnu jsou ze začátku pracovního týdne vyšší, pak klesají do pátku a o víkendu se tyto hodnoty zase zvednou nad svůj týdenní průměr. Tento vzor se opakuje do vypuknutí první vlny pandemie, ta poté tento vzor změní na pravý opak. To znamená, že nejvyšší hodnoty jsou dosaženy přes týden, tedy postupně stoupají od pondělí do čtvrtka, v pátek hodnoty klesnou na nižší hodnotu, a poté se hodnota pomalu začne zvyšovat o víkendu.

Po pozorování jsem se rozhodl udělat dekompozici časové řady,tedy rozložit časovou řadu na trend, sezónní, a reziduální složku. Trendová složka je vytvořená pomocí metody sezónní dekompozice v knihovně Statsmodels a představuje trend vytvořený pomocí metody klouzavých průměrů. Na obrázku 4.1 můžeme vidět 2 křivky, modrá křivka značí nasbíraná data a oranžová křivka značí trend. Na osách grafu se nachází datum naměřených hodnot a počet připojených uživatelů. V březnu roku 2020 si můžeme na grafu všimnout jistého zvýšení hodnot, toto zvýšení souvisí s první vlnou Covidové pandemie, která nejsilněji ovlivnila počet připojených uživatelů. Tyto vlny lze pozorovat i nadále ke konci roku 2020 a začátkem roku 2021. Z grafu si lze povšimnout téže sezónního opakování našich hodnot. Toto opakování je vždy týdenní, nejmenší hodnoty jsou pravidelně v pátek, odkud začínají růst a pomalu se zvětšují do prostředka následujícího týdne.

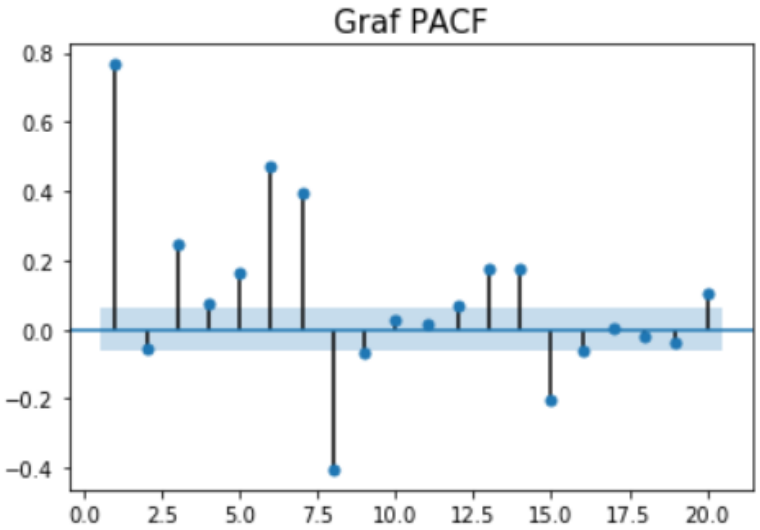


Obrázek 4.1: Trend celé časové řady vytvořený knihovnou Statsmodels

Po prozkoumání jednotlivých složek jsem se rozhodl statisticky otestovat stacionaritu naší časové řady pomocí Dickey-Fuller testu. Hodnota p vyšla 0.032. To znamená, že je naše časová řada připravená na vytvoření prediktivních modelů.

4.2.3 Prediktivní model AR

Po zjištění, že má data jsou stacionární, jsem se rozhodl vytvořit první prediktivní model. Tím je model AR, který predikuje hodnoty na základě předchozích hodnot v časové řadě. Před vytvořením modelu jsem si nejprve vytvořil graf PACF, abych zjistil, jaká úroveň zpoždění se pro moje data bude nejvíce hodit. Na obrázku 4.2 se nachází graf s PACF naší časové řady s výší zpoždění 20. Na ose x je výše daného zpoždění a na ose y je hodnota, jak dané zpoždění ovlivňuje predikovanou hodnotu v intervalu od -1 do 1. Z obrázku můžeme vyčíst, že nejzajímavější pro nás budou hodnoty 1, 3, 5, 6 a 8.



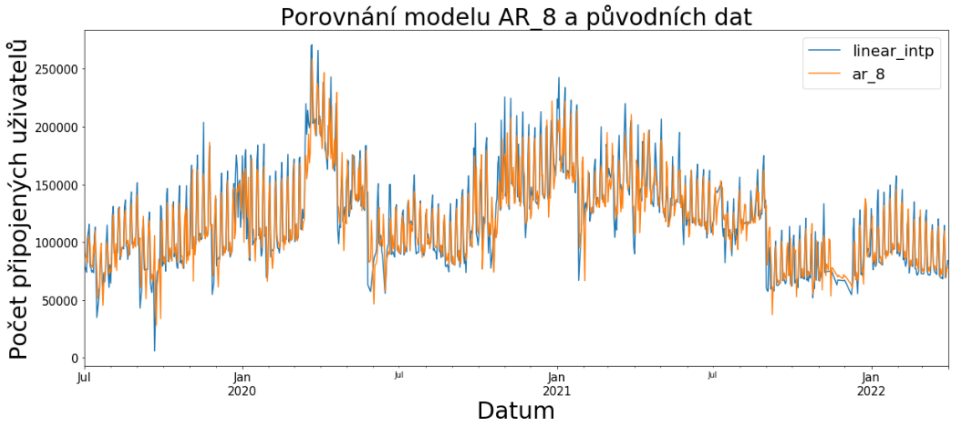
Obrázek 4.2: Graf PACF pro naměřená data

Poté jsem se rozhodl postupně vytvořit co nejlepší model AR(x), který by měl predikovat hodnoty dat. AR modely mají tu vlastnost, že čím vyšší číslo zpoždění x, tím více by měli být přesnější. To ale nemusí být vždy pravda, protože si sebou do výsledné hodnoty přidáváme vliv, jak podstatných hodnot pro naši predikci, tak i vliv nepodstatných hodnot. Rozhodl jsem se tedy pro trénování modelu postupně od AR(1) až po AR(x), kde x bude poslední model se zpožděním takovým, že se nebude lišit výsledek jeho LLR testu od minulého modelu, a zároveň bude pořád dostatečně ovlivňovat výslednou hodnotu. Pro každý model jsem si vygeneroval tabulku pomocí knihovny Statsmodels, ze které se dají pro každý model vyčíst hodnoty:

- **AIC, BIC, HQIC** jsou statistická kritéria pro vybírání modelu. Při výběru modelu se snažíme tyto hodnoty minimalizovat.
- **Metoda maximální věrohodnosti** značí jak moc je pravděpodobné, že se náhodná veličina předpoví správně hledanou hodnotu. Nejlepší model bude mít tuto hodnotu nejvyšší.

- **koeficient** uvádí s jakým koeficientem c a koeficienty ϕ se vytvořil náš model.
- **std error** ukazuje, jak se průměrně odchyluje model od naměřených dat.
- **z** je podíl hodnoty koeficientu a erroru, nazýváme ho standardizovaný koeficient
- **P>|z|** Tento sloupec nám říká jakou hodnotu p má daný koeficient a pokud je tato hodnota vyšší než 0.05, což je naše daná hranice 5 procent, může model predikovat nepřesné výsledky.

Z testu PACF si můžeme všimnout, že nejvyšší podstatná hodnota pro aktuální hodnotu v naší časové řadě je hodnota se zpožděním 8. Rozhodl jsem se ještě pro jistotu vytvořit model AR(9), abych zjistil, že model opravdu již nevylepšuje predikci a vybral jsem AR(8) jako nejlepší model z autoregresních modelů. Na obrázku 4.3 můžeme vidět graf porovnání naměřených hodnot doplněných o lineární interpolaci jako modrou křivku a graf predikce pomocí funkce AR(8) jako oranžovou křivku. Z grafu si lze povšimnout, jak prediktivní model pochytil trend námi naměřených hodnot, ale když se vyskytne v grafu nepředvídatelně vysoká nebo nízká hodnota, tak si s ni model moc neumí poradit.



Obrázek 4.3: Graf porovnání modelu AR(8) a naměřených dat

4.2.4 Prediktivní model AR na krátkých datech

Po předchozím modelování mě zajímalo, jak moc ovlivňují data doplněná o lineární interpolaci predikci chybějících dat. Rozhodl jsem se vytvořit prediktivní model časové řady na datech neobsahující mezery. Tím je interval 29.11.2019 do 26.5.2020, který je nejdelší spojitou částí naměřených dat. Zvolil jsem zde stejný přístup k analýze jako v předchozím případě, vytvořil jsem si tedy sezónní dekompozici časové řady, a následně jsem provedl Dickey-fuller test. Z testu vyšlo najevo, že řada není slabě stacionární, a

proto jsem zkoušel metody, jakými bych mohl řadu stacionární udělat. Nejprve jsem zkusil diferencování. Při diferencování se každá hodnota funkce změní o rozdíl s hodnotou předchozí. Se získáním nových hodnot jsem zkusil znovu Dickey-Fuller test. Ten mi znovu naznačil, že i nová časová řada není stacionární. Hledal jsem tedy jinou metodu pro stacionarizaci řady. Vybral jsem si eliminování trendu vypočítaného v předchozím kroku. Po eliminování trendu vyšla hodnota p po zpracování testu 0.000004 což znamená, že je řada opět stacionární. Poté jsem znovu vytvořil graf PACF, podle kterého jsem se rozhodoval, jaké zpoždění bude maximální pro trénování mého modelu. Zjistil jsem, že nejlépe se znovu bude hodit model AR(8). Vytvořil jsem tedy i predikci tohoto modelu. Vytvořený model se téměř schoduje s modelem AR(8) na celé časové řadě. Proto jsem se rozhodl použít složitější model.

4.2.5 Prediktivní model ARMA

První dva prediktivní modely se sobě velmi podobají, tím pádem jsem se rozhodl pro další pozorování použít metodu ARMA na zkrácených datech. ARMA kombinuje metodu AR a metodu MA, neboli metodu klouzavých průměrů. Pro vytvoření funkce jsem následoval stejný postup jako při vytvoření funkce AR. Tedy eliminoval jsem trendovou složku, a následně jsem hledal nejlepší model AR. Ten jsem už v předchozím bodě zjistil, že jím je funkce AR(8). Poté jsem téměř identický postup použil pro MA modely, z něhož jsem zjistil, že ze skupiny modelů MA, jsou pro predikování hodnot na mojí časové řadě nejlepší modely MA(3) a MA(5). Dalším krokem bylo nasadit kombinovaný model ARMA. Tento model je kombinací modelů AR a MA. Nejprve jsem vložil model ARMA(1,1), abych měl porovnání. A následně jsem vložil model ARMA (8,3), který vychází z modelů zjištěných v předchozím kroku. Ten byl pro moje počítání ale zbytečně složitý a při porovnávání modelů jsem zjistil že model ARMA(5,3) má velmi podobné výsledky jako model ARMA (8,3). Při pozorování modelu ARMA (5,3) jsem zjistil, že počet nesignifikantních hodnot je příliš velký. Zkusil jsem proto použít podobný model a tedy model ARMA (5,2). Jeho výsledek nám ukazuje pouze 2 nesignifikantní hodnoty, a to u MA(2) a AR(3). Z toho důvodu jsem se pro následnou predikci rozhodl využít tento model.

4.3 Porovnání

Na konec jsem se rozhodl vytvořené modely porovnat v tom, jak jsou schopné predikovat hodnoty na časové řadě. Tyto modely jsem se rozhodl graficky porovnat s původními daty a následně určit, který model se pro predikci dat hodí nejlépe.

Pro porovnání hodnot jsem si vytvořil tabulku hodnot 4.1, která obsahuje porovnání s původními daty. Pro toto porovnání jsem znovu vybral nejdelší spojitý úsek, z něj jsem si vybral prosinec roku 2019. V prvním sloupci tabulky se nachází datum provedeného měření, ve druhém sloupci se nachází naměřené hodnoty, ve třetím sloupci je predikce prvního modelu AR(8), který

je vytvořený pomocí lineární interpolace původních dat a v posledním sloupci je model ARMA(5,2), který jsem vytvořil eliminací trendu na zkrácených datech.

Tabulka 4.1: Porovnání modelů AR(8) a ARMA (5,2) s původními daty

Datum	Původní data	AR(8)	ARMA(5,2)
02.12.2019	79552	105838	111952
03.12.2019	85056	79200	75865
04.12.2019	85424	61119	97238
05.12.2019	86924	81206	79985
06.12.2019	88142	86640	93811
07.12.2019	141260	124346	118593
08.12.2019	159818	143349	131072
09.12.2019	88918	109654	111498
10.12.2019	91076	88401	85471
11.12.2019	91014	96540	101956
12.12.2019	92376	90650	79587
13.12.2019	94056	94493	93883
14.12.2019	148660	135140	138216
15.12.2019	161562	153425	145392
16.12.2019	92548	107767	111484
17.12.2019	94196	91443	90926
18.12.2019	87120	99912	108614
19.12.2019	91074	89157	79795
20.12.2019	92060	96366	92497
21.12.2019	142626	135857	138678
22.12.2019	150878	147538	145284
23.12.2019	115416	102979	110717
24.12.2019	155030	107747	109802
25.12.2019	175426	135527	153656
26.12.2019	165886	146446	139220
27.12.2019	121530	143374	128817
28.12.2019	151080	150018	136992
29.12.2019	153658	151433	143753
30.12.2019	112828	127046	130666
31.12.2019	130910	140048	137797

V tabulce 4.1 je vidět týdenní rozložení hodnot. Nejvyšší týdenní hodnoty se nacházejí ve středeční nebo čtvrteční dny, a následně hodnoty klesají k víkendu, kde začnou znovu růst. Dále je v tabulce vidět jak sváteční dny ovlivnili naše hodnoty, a to silným navýšením připojených uživatelů ve dnech volna 24.12, 25.12. a 26. 12. V tyto dny prediktivní modely nedokázali odhadnout tak náhlý nárůst, a proto lze pozorovat, že data predikována v tyto dny jsou nepřesná a výrazně nižší než data naměřená. Tyto nepřesnosti následně ovlivnily modely natolik, že následující dny jsou předpovězené velice dobře.

Při pohledu na tabulku lze pozorovat, že predikce vytvořená samotným autoregresním modelem se blíží více našim původním hodnotám. Tato skutečnost je daná tím, že model ARMA není na predikci této řady optimální. Model je vytvořen pomocí detrendované řady, která je sice stacionární, ale není úplně reziduální. Pro model MA(x) je důležité, aby časová řada, na které je model postavený, byla dokonalý bílý šum. To se bohužel u mého modelu nepodařilo úplně. S tímto zjištěním jsem tedy označil model AR(8) za nejlepší prediktivní model.

Kapitola 5

Závěr

Velká data jsou široký pojem, pod kterým si můžeme představit data získaná například ze sociálních sítí, klientských databází a nebo e-shopů. Jejich využití a analýza je pro většinu firem významnou prioritou, protože je dokáží využít k inovaci a zlepšení konkurenceschopnosti. Tato práce se věnuje takovým datům a postupům jejich analýzy. Zvolená data jsou shromážděna z webové stránky, konkrétně se jedná o denní počet připojených uživatelů v průběhu necelých tří let.

V první části definuji sesbíraná data a zároveň přiblížím postupy jejich analýzy. Tato část působí jako teoretický základ, který je následně rozvinut představením modelů predikce hodnot se zaměřením na teorii časových řad. Tento teoretický rámec pak využiji v praktické části, při implementaci modelů a porovnání jejich výstupů.

Tato práce měla za cíl analyzovat časovou řadu a pomocí teorie časových řad vytvořit prediktivní model a tento model následně použít na predikci chybějících dat v naměřených datech.

Na základě definované teorie, využitých modelů a porovnaných výsledků lze usoudit, že cíl práce byl naplněn, data byla analyzována a následně byl vytvořen prediktivní model časové řady. Model AR(8) vytvořený na zinterpolované časové řadě má nejlepší vlastnosti pro predikci ze všech porovnávaných modelů. Pomocí tohoto modelu jsme schopni predikovat chybějící hodnoty v minulosti a zároveň model může predikovat i budoucí hodnoty, které ještě nebyly naměřeny.

5.1 Budoucnost predikce

Při dalším modelování časové řady bych použil složitější predikativní modely, například model SARIMA, který se může naučit i predikci na základě sezónní složky naměřené časové řady. Pro takový model bych doporučil zapracovat predikci na základě českého kalendáře. Použití kalendáře doporučuji, protože naměřené hodnoty se velmi liší o víkendech, nebo o vánočních svátcích. K tomuto modelu bych přidal metodu hyperparametru.

Hyperparametr[19] modelu je charakteristika modelu, která je vůči modelu externí a jejíž hodnotu nelze odhadnout z dat. Hodnota hyperparametru musí být nastavena před zahájením procesu strojového učení. Pro hledání hyperparametru bych doporučil metodu Grid search. Pomocí tohoto modelu bychom mohli dosáhnout přesnějších hodnot predikce než jsou pomocí modelu AR(8).

Literatura

[1] Doug Laney, *3d data management: Controlling data volume, velocity and variety*, META Group Research Note 6 (2001).

[2] *Vše o koronaviru. Počty nakažených, statistiky, praktické rady i hlasy odborníků - Aktuálně.cz. Zprávy - Aktuálně.cz* [online]. Copyright © [cit. 22.08.2022]. Dostupné z: https://zpravy.aktualne.cz/widget-koronavirus-vse-o-koronaviru/r 28c952346d0611eaa6f6ac1f6b220ee8/

[3] Anaconda Navigator — *Anaconda documentation. Anaconda Documentation — Anaconda documentation* [online]. Dostupné z: https://docs.anaconda.com/anaconda/navigator/

[4] Conda — *conda documentation.* [online]. Copyright © Copyright 2017, Anaconda, Inc. Dostupné z: https://docs.conda.io/en/latest/

[5] Jupyter Project Documentation — *Jupyter Documentation 4.1.1 alpha documentation.* . Copyright © Copyright 2015, Jupyter Team. Dostupné z: https://docs.jupyter.org/en/latest/

[6] Our Documentation | *Python.org. Welcome to Python.org . Copyright ©2001* . Dostupné z: https://www.python.org/doc/

[7] NumPy Documentation. *NumPy. Copyright © Copyright 2008.* Dostupné z: https://numpy.org/doc/

[8] pandas documentation — pandas 1.4.3 documentation. pandas - *Python Data Analysis Library . Copyright © Copyright 2008* . Dostupné z: https://pandas.pydata.org/docs/

[9] Matplotlib documentation — *Matplotlib 3.5.2 documentation. Matplotlib — Visualization with Python. Copyright © Copyright 2002.* Dostupné z: https://matplotlib.org/stable/index.html

[10] Seabold, S., Perktold, J. (2010). *statsmodels: Econometric and statistical modeling with python.* In 9th Python in Science Conference.

[11] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... SciPy 1.0 Contributors. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.* Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

[12] *Linear Interpolation Formula - Derivation, Formulas, Examples. Online Math Classes* | Cuemath . Dostupné z: https://www.cuemath.com/linear-interpolation-formula/

[13] KŘIVÝ, Ivan. *Analýza časových řad* [online]. [cit. 2015-09-18]. Dostupné z: https://publi.cz/download/publication/20?online=1

[14] *Časové řady.* Analýza časových řad. (n.d.). Retrieved September 8, 2022, from https://iastat.vse.cz/casovky/casovky2.htm

[15] 8.3 Autoregressive models | *Forecasting: Principles and Practice (2nd ed).* OTexts. Dostupné z: https://otexts.com/fpp2/AR.html

[16] Autoregressive moving average (ARMA) - explained. *The Business Professor,* LLC. (n.d.), Dostupné z: https://thebusinessprofessor.com/en-US/research-analysis-decision-science/autoregressive-moving-average-arma-definition

[17] Fox, J. (1997) *Applied regression analysis, linear models, and related methods.* Thousand Oaks, CA: Sage Publications.

[18] *Interpret all statistics and graphs for Augmented Dickey-Fuller test. Minitab.* (n.d.). Dostupné z: https://support.minitab.com/en-us/minitab/21/help-and-how-to/statistical-modeling/time-series/how-to/augmented-dickey-fuller-test/interpret-the-results/all-statistics-and-graphs/

[19] Joseph, R. (2018, December 29). *Grid search for model tuning.*, Dostupné z: https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e