DELTA - Střední škola informatiky a ekonomie, s.r.o.
Ke Kamenci 151, Pardubice

Využití metod strojového učení pro analýzu vitálních dat

Tereza Radmila Šulcová
4.A
Informační technologie (18-20-M/01)

2024/2025

Poděkování

Chci poděkovat Ing. Richardu Cimlerovi, Ph.D. a Ing. Daliboru Cimrovi, Ph.D. za vedení při tvorbě práce a možnost pracovat na tomto projektu.

Anotace

Cílem projektu je využít klastrovou analýzu pro rozdělení dat podle trendů uživatele. Pomocí neuronových sítí klasifikovat uživatele do skupin na základě jejich aktivity. Zhodnotit vliv velikosti vstupního datasetu na přesnost klasifikace neuronových sítí.

Klíčová slova

strojové učení, klastrová analýza, neuronové sítě, předzpracování dat, klasifikace uživatelů, cirkadiánní rytmus

Annotation

The aim of the project is to utilize cluster analysis for categorizing data based on user trends. Neural networks will be used to classify users into groups according to their activity patterns. Additionally, the project will assess the impact of the input dataset size on the classification accuracy of neural networks.

Keywords

machine learning, cluster analysis, neural networks, data preprocessing, user classification, circadian rhythm

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení:	Tereza Šulcová
Školní rok:	2024/2025
Třída:	4. A
Obor:	Informační technologie 18-20-M/01

Téma práce:	Využití metod strojového učení pro analýzu vitálních dat
Vedoucí práce:	Ing. Richard Cimler, Ph.D.

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:
Cílem práce je zhodnotit různé metody strojového učení pro analýzu vitálních dat lidí či zvířat. Pro zhodnocení budou využity stávající datové sady. Student se během vypracovávání závěrečné práce seznámí s procesem čištění a preprocesingu dat a s vybranými metodami strojového učení pro analýzu rozsáhlých datových souborů.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):
září, říjen:

- nabytí teoretických znalostí potřebných k realizaci
- rešerše existujících prací
- průzkum dostupných nástrojů a modelů

listopad:

- čištění a preprocessing dat
- experimentace s využitím vybraných metod strojového učení

prosinec, leden:

- zhodnocení vybraných metod

únor, březen:

- interpretace a porovnání výsledků
- sepsání výsledků ve formě závěrečné práce

Prohlašuji, že jsem maturitní projekt vypracovala samostatně, výhradně s použitím uvedené literatury.

V Pardubicích, 27.3.2025

Tereza Radmila Šulcová

Obsah

1. Úvod

1.1 Cirkadiánní rytmus

Tento rytmus je biologickým rytmem s periodou 20-28 hodin. Patří mezi biorytmy, které jsou charakterizovány kolísáním aktivit a bdělosti, nejčastěji s denní, měsíční nebo roční periodou. Název tohoto rytmu pochází z latiny: "circa" znamená "okolo" nebo "během" a "dies" znamená "den". Konceptu se jako první věnoval rumunský vědec Franz Halberg, ovšem jev samotný je znám již od starověku. [1]

2. Použité metody

2.1 Základy klastrové analýzy

Klastrová analýza je statistická metoda používaná pro seskupování datových bodů do klastrů (clusterů) tak, aby prvky uvnitř jednoho shluku byly co nejpodobnější, zatímco prvky mezi shluky byly co nejvíce odlišné. Používá se v mnoha oblastech, jako je rozpoznávání vzorů, bioinformatika, strojové učení nebo analýza obrazu. [2]

2.1.1 Předzpracování dat (data preprocessing)

Předzpracování se týká transformací, které se na naše data aplikují před jejich předáním algoritmu. Předzpracování dat je technika, která se používá k převodu surových dat na čistý soubor dat. Pro různé modely musí být data v různých formátech. Např. model Random forest není schopen zpracovat data, která obsahují NaN hodnoty, a tak je třeba se jich zbavit, nastavením hodnoty na 0, průměrem okolních hodnot nebo dalšími technikami. [3]

2.1.2 Analýza hlavních komponent (principal component analysis)

Techniku analýzy hlavních komponent zavedl matematik Karl Pearson v roce 1901. Funguje na základě podmínky, že zatímco data ve vyšším rozměrovém prostoru jsou mapována na data v nižším rozměrovém prostoru, rozptyl dat v nižším rozměrovém prostoru by měl být maximální. Tato analýza je technika pro algoritmy patřící do učení bez učitele, která se používá ke zkoumání vzájemných vztahů mezi souborem proměnných. Je také známá jako obecná faktorová analýza, kde regrese určuje přímku nejlepší shody. Používá ke snížení dimenzionality souboru dat nalezením nového souboru proměnných, který je menší než původní soubor proměnných, zachovává většinu informací o vzorku a je užitečný pro regresi a klasifikaci dat. [4]

2.1.3 Učení bez učitele (unsupervised learning)

Učení bez učitele je trénování stroje pomocí dat, která nejsou klasifikovaná ani označená, a umožňuje algoritmu jednat na základě těchto informací bez vedení. Úkolem stroje je zde seskupovat netříděné informace podle podobností, vzorů a rozdílů bez předchozího trénování dat. Toto trénování se dělí do dvou kategorií. Shlukování (clustering), při kterém se zjišťují skupiny obsaženy v datech. Například tato skupina zákazníků nakupuje převážně výrobek X a

tato výrobek Y. Asociace je druhá metoda, jež je založena na hledání asociací mezi velkou částí dat. Například zákazníci, kteří nakupují výrobek X mají tendenci nakupovat výrobek Y. [5] [6]

2.1.4 Učení s učitelem (supervised learning)

Učení s učitelem je proces, při kterém trénovací data obsahují vstupní objekty a jejich požadovaná výstupní ohodnocení, tj. výrok učitele o objektu. Tento proces využívá zpětnou vazbu podobně jako biologické sítě. Neuronové síti je předložen vzor, a na základě aktuálního nastavení sítě je zjištěn výsledek. Ten je porovnán s požadovaným výsledkem a určena chyba. Poté je dle typu neuronové sítě spočítána nutná korekce a upraveny hodnoty vah nebo prahů, aby se snížila hodnota chyby. Tento proces se opakuje až do dosažení stanovené minimální chyby, po které je síť adaptována.

Výstupem naučené funkce mohou být buď spojité hodnoty (při regresi), nebo binární hodnoty označující příslušnost vstupních objektů do daných tříd (při klasifikaci). Naučená funkce je schopna odhadovat výstupní ohodnocení každého vstupního objektu poté, co zpracuje trénovací příklady. To vyžaduje, aby síť dokázala generalizovat souvislost mezi vstupy a výstupy na základě příkladů obsažených v trénovacích datech. [7]

2.2 Druhy algoritmů

2.2.1 K-means algoritmus

K-means algoritmus je součástí učení bez vedení. Má jeden parametr, celé číslo počet shluků, který uživatel určuje před spuštěním algoritmu na datech. Jeho výchozí hodnota je 8. Algoritmus na datech funguje tak, že se náhodně vybere tolik bodů v datech, kolik je požadovaný počet klastrů, které budou fungovat jako centroidy klastrů. Poté se pro všechny centroidy zvlášť střídají dva kroky. První, kde se vypočítá vzdálenost všech bodů od daného centroidu, přičemž se ten nejbližší přiřadí k shluku. A druhý, při kterém se vypočítá nový centroid, který je průměrem všech bodů v shluku. Tyto dva kroky se opakují dokud se shluk neustálí, neboli centroid nemění svou pozici. Takto se algoritmus opakuje dokud nejsou všechny body v nějakém klastru. Časová složitost je O(n*k*i*d) a paměťová složitost je O(n*d), kde n je počet bodů, k je počet klastrů, i je počet iterací, které algoritmus udělá, dokud centroidy nejsou stabilní a d je počet dimenzí, které data mají. [8] [9] [10] [11]

2.2.2 Hierarchické klastrování (hierarchical clustering)

Hierarchické shlukování je druh shlukování založený na konektivitě, která seskupuje datové body, které jsou si blízké na základě míry podobnosti nebo vzdálenosti. Předpokladem je, že datové body, jež jsou si blízké, jsou si podobnější nebo příbuznější než datové body, které jsou

od sebe vzdálenější. Hierarchické vztahy mezi skupinami znázorňuje dendrogram. Jednotlivé datové body jsou umístěny ve spodní části, zatímco největší shluky, zahrnující všechny datové body, jsou umístěny nahoře. Aby bylo možné vytvořit libovolný počet klastrů, lze dendrogram rozřezat v jakékoli výšky. Jsou dva druhy tohoto shlukování. První se nazývá divisivní a funguje na principu tzv. "shora dolů". Všechna pozorování začínají v jednom shluku a rozdělení na více menších shluků se rekurzivně provádí tak, jak se v hierarchii postupuje dolů. Druhý druh je agglomerativní a jeho fungování je popsáno v další kapitole. [12] [13]

2.2.2.1 Hierarchické agglomerativní klastrování (hierarchical agglomerative clustering)

Tento druh hierarchického shlukování funguje na principu tzv. "zdola nahoru". Pozorování vždy začíná v jednom shluku a poté se dvojice klastrů, které jsou vzdáleností či hodnotou nejblíže, spojují, jak se postupuje nahoru v hierarchii. Stejně jako K-means algoritmus přijímá celé číslo jako parametr pro určení počtu shluků. Jeho výchozí hodnota je 2. Časová složitost tohoto algoritmu je O(n3) a paměťová složitost je O(n2), kde n je počet bodů. [12] [13]

2.2.3 DBSCAN

Density-based spatial clustering of applications with noise, zkráceně DBSCAN, je algoritmus, který na datasetu do shluků dává body, které mají kolem sebe hodně "sousedů" a body, které jsou naopak samy, označí jako šum (noise). Na začátku algoritmus najde vzorky bodů, jež budou fungovat jako centroidy shluků. Proto je dobré když mají data klastry s podobnou hustotou bodů. Pro tento algoritmus se nastavují dva parametry a to eps a min samples. Eps je desetinným číslem, které určuje maximální vzdálenost mezi dvěma body pro které stále platí, že jsou "sousedé" a patří do jednoho klastru. Výchozí hodnota je 0,5. Druhý parametr, min samples, je celé číslo, které určuje počet "sousedů", které bod musí mít aby byl vyhodnocen jako centroid. Toto číslo vztahuje do parametru i samotný centroid. Čím je jeho hodnota vyšší tím hustší shluky algoritmus hledá. Jeho výchozí hodnota je 5. Nejhorší časová náročnost je O(n2), jež se může objevit, když je eps parametr velký a min samples je malý. HDBSCAN je verze DBSCAN, která provádí stejný algoritmus, avšak ho provádí nad různými hodnotami eps a integruje výsledek tak, aby našel klastrování, jež má nejlepší stabilitu v závislosti na eps. Tento algoritmus může lépe hledat shluky s různou hustotou oproti DBSCAN. [14] [15] [16]

2.3 Evaluace výsledků algoritmů

Při použití shlukovacích technik je velmi důležité vyhodnotit kvalitu vytvořených shluků. Tyto metriky jsou kvantitativní ukazatele používané k hodnocení výkonnosti a kvality klastrovacích algoritmů. Mezi populární druhy metrik patří interní a externí evaluace (viz vnitřní evaluace a vnější evaluace). [17]

2.3.1 Vnitřní (internal) evaluace

Pro tento druh metrik je výsledek vyhodnocen na základě dat, která byla shlukována. Tyto metriky obvykle přiřazují nejlepší skóre algoritmu, který vytváří shluky s vysokou podobností uvnitř klastru a nízkou podobností mezi shluky. Jednou z nevýhod je, že vysoké skóre v interním měřítku nemusí nutně vést k efektivním aplikacím pro vyhledávání informací. Navíc je toto hodnocení zaujaté vůči algoritmům, které používají stejný model shluku. Například shlukování K-means přirozeně optimalizuje vzdálenosti objektů a interní kritérium založené na vzdálenosti pravděpodobně nadhodnotí výsledné klastrování. [17]

2.3.1.1 Silhouette score

Tato technika poskytuje stručné grafické znázornění toho, jak dobře byl každý objekt klasifikován. Navrhl ji belgický statistik Peter Rousseeuw v roce 1987. Hodnota siluety je měřítkem toho, jak je objekt podobný svému vlastnímu shluku (koheze) ve srovnání s ostatními shluky (separace). Pohybuje se od -1 (špatné klastrování) do +1 (dokonalé shlukování). Skóre blízké 1 naznačuje dobře oddělené shluky. Toto skóre je užitečné pro zjištění, jak vhodné jsou metody shlukování a kolik shluků je pro určitý soubor dat nejvhodnější. [18]

2.3.1.2 Davies-Bouldin index

Davies-Bouldin index se používá pro hodnocení účinnosti klastrovacích algoritmů. Hodnotí kompaktnost a oddělení shluků datového souboru. Tato metrika byla uvedena v roce 1979 David L. Daviesem a Donald W. Bouldinem. Lépe definované shluky jsou označeny nižším indexem, který se určuje porovnáním průměrného poměru podobnosti a nepodobnosti každého shluku s poměrem podobnosti a nepodobnosti jeho nejpodobnějšího souseda. Protože shluky s nejmenšími vnitroshlukovými a největšími mezishlukovými vzdálenostmi poskytují nižší index, pomáhá to při určování ideálního počtu shluků. [19]

2.3.1.3 Calinski-Harabasz index

K hodnocení kvality shluků v rámci souboru dat se používá Calinski-Harabaszův index také známý jako Variance Ratio Criterion (VRC). Představili ho Tadeusz Caliński a Jerzy Harabasz v roce 1974. Vyšší hodnoty ukazují na kompaktní a dobře oddělené shluky. Vypočítává poměr rozptylu uvnitř shluku a rozptylu mezi shluky. Pomáhá určit ideální počet klastrů pro daný soubor dat porovnáním indexu v různých shlucích. Lepší vymezení shluků je implikováno vyšším Calinski-Harabaszovým indexem. Tato míra je užitečná pro posouzení toho, jak dobře fungují klastrovací algoritmy, což pomáhá vybrat nejlepší řešení shlukování pro různé datové sady. [20]

2.3.2 Vnější (external) evaluace

Při externí evaluaci se výsledky shlukování hodnotí na základě dat, která nebyla použita pro shlukování, například známých značek tříd a externích srovnávacích testů. Taková měřítka se skládají ze souboru předem klasifikovaných položek a tyto soubory často vytvářejí lidé. [17]

2.3.2.1 Rand index

Rand index, pojmenován po Williamu M. Randovi, vypočítává, jak jsou shluky získané shlukovacím algoritmem podobné referenčním klasifikacím. Jedním z problémů Rand indexu je, že falešně pozitivní a falešně negativní výsledky mají stejnou váhu. To může být pro některé klastrovací aplikace nežádoucí vlastnost. F-measure tento problém řeší, stejně jako Adjusted Rand Index (ARI), který vykonává Rand index a bere v úvahu očekávanou podobnost mezi dvěma náhodnými klastry stejných dat. [21] [22]

2.3.2.2 F-measure

Předpokládá se, že název F-measure byl pojmenován podle jiné funkce F ve Van Rijsbergenově knize, když byla představena na the Fourth Message Understanding Conference. Výpočet metriky se skládá z přesnosti a zpětné vazby testu. Přesnost je počet pravdivě pozitivních výsledků vydělený počtem všech vzorků, které měly být pozitivní, včetně těch, které nebyly správně identifikovány. Zpětná vazba je počet pravdivě pozitivních výsledků vydělený počtem všech vzorků, které měly být identifikovány jako pozitivní. [23]

2.4 Neuronové sítě

Neuronová síť je model inspirovaný strukturou a funkcí biologických neuronových sítí v mozcích zvířat. Skládá se ze spojených jednotek nebo uzlů zvaných umělé neurony, které volně modelují neurony v mozku. Ty jsou propojeny hranami, které modelují synapse v mozku. Každý umělý neuron přijímá signály od připojených neuronů, poté je zpracovává a vysílá signál dalším připojeným neuronům. „Signál“ je reálné číslo a výstup každého neuronu se vypočítá pomocí určité nelineární funkce součtu jeho vstupů, která se nazývá aktivační funkce. Síla signálu na každém spojení je určena váhou, která se upravuje během procesu učení.

Neurony se obvykle sdružují do vrstev. Různé vrstvy mohou na svých vstupech provádět různé transformace. Signály putují z první vrstvy, tzv. vstupní vrstva, do poslední vrstvy, tzv. výstupní vrstva, přičemž mohou procházet několika mezivrstvami, tzv. skrytými vrstvami. Síť se nazývá hluboká neuronová síť, pokud má alespoň dvě skryté vrstvy. [24]

2.4.1 Historie neuronové sítě

První umělé neurony vytvořili Warren McCulloch a Walter Pitts v roce 1943. Fungovaly na principu binárního výstupu 0/1 podle prahové hodnoty vážené sumy vstupů. Jejich teorie byla, že dokáží počítat aritmetické a logické funkce, avšak nebyla pro ně vypracována žádná tréninková metoda. V padesátých letech přichází Frank Rosenblatt s perceptony. Percepton je nejjednodušší model dopředné neuronové sítě (Feed-forward Neural Network) s učením s učitelem (supervised learning). Perceptrony využívá k rozpoznávání vzorů. Mimo to dokazuje, že pokud existují váhy, které zadaný problém řeší, pak k nim učící pravidlo konverguje. Počáteční nadšení však uvadá, když je zjištěno, že takovýto perceptron umí řešit pouze lineárně separovatelné úlohy. Frank Rosenblatt se sice snaží model upravit a rozšířit, ale nedaří se mu to a tak až do osmdesátých let přestává být o perceptrony a neuronové sítě zájem. V 80. letech přišel průlom s vícevrstvými perceptronovými sítěmi schopnými aproximovat libovolnou vektorovou funkci. Zájem později klesl kvůli problémům s učením hlubokých sítí. Nejužívanější metoda učení je algoritmus zpětného šíření chyby, odvozený z dynamického programování. K jeho vývoji přispěli Kelley (1960), Bryson (1961) a Dreyfus (1962). Linnainmaa (1970) publikoval obecnou metodu automatického derivování. Werbos (1974, 1982) aplikoval princip na neuronové sítě. Rumelhart, Hinton a Williams (1986) prokázali užitečnost pro hluboké učení. V současnosti se neuronové sítě převážně užívají v úlohách zpracovávání přirozeného jazyka (natural language processing v rámci tzv. jazykových modelů jako např. Word2Vec nebo Transformer a v úlohách počítačového vidění (computer vision) v rámci tzv. konvolučních neuronových sítí. [25] [26] [24]

2.4.2 Druhy architektur neuronových sítí

2.4.2.1 Dopředná neuronová síť (Feed-forward Neural Network)

Dopředná neuronová síť je charakteristická jednosměrným tokem informací mezi jednotlivými vrstvami sítě. Tok informací jde pouze dopředu od vstupní vrstvy neuronů, pokud existují přes skryté vrstvy neuronů, až k výstupní vrstvě neuronů, tj. bez jakýchkoli smyček či zpětných vazeb. [27]

2.4.2.2 Rekurentní neuronová síť (Recurrent Neural Network)

Rekurentní neuronová síť se vyznačují směrem toku informací mezi jednotlivými vrstvami sítě. Tok informací v rekurentní síti je obousměrný, tj. tvoří se smyčky. Zpracovávají data ve více časových krocích, a díky tomu jsou dobře uzpůsobeny pro modelování a zpracování textu, řeči a časových řad.

Základním stavebním kamenem RNN je rekurentní jednotka. Tato jednotka udržuje skrytý stav, v podstatě jakousi formu paměti, která je v každém časovém kroku aktualizována na zák-

ladě aktuálního vstupu a předchozího skrytého stavu. Tato zpětnovazební smyčka umožňuje síti učit se z minulých vstupů a zahrnout tyto znalosti do aktuálního zpracování.

První sítě RNN trpěly problémem mizejícího gradientu, což omezovalo jejich schopnost učit se závislosti na dlouhé vzdálenosti. Tento problém vyřešila varianta dlouhé krátkodobé paměti (LSTM) v roce 1997, a stala se tak standardní architekturou RNN.

RNN se uplatnily v úlohách, jako je nesegmentované, propojené rozpoznávání rukopisu, rozpoznávání řeči, zpracování přirozeného jazyka a neuronový strojový překlad. [28]

2.4.3 Učení neuronových sítí

Cílem učení neuronové sítě je nastavit síť tak, aby dávala co nejpřesnější výsledky.
V biologických sítích jsou zkušenosti uloženy v synapsích. V umělých neuronových sítích jsou zkušenosti uloženy v jejich matematickém ekvivalentu – váhách. Učení neuronové sítě rozlišujeme na učení s učitelem (supervised learning) a učení bez učitele (unsupervised learning) (viz kapitoly učení s učitelem, učení bez učitele). U neuronových sítí se tyto dva hlavní druhy učení dělí na další. Fáze učení neuronové sítě bývá nazývána adaptivní, fáze vybavování po naučení neuronové sítě bývá nazývána aktivní. [25]

2.4.3.1 Hebbovské učení

Hebbův princip představil v roce 1949 Donald Olding Hebb. Hebbovské učení je metodou určování, jak měnit váhy mezi umělými neurony. Váha mezi dvěma neurony se zvyšuje, pokud se oba neurony aktivují současně, a snižuje, pokud se aktivují odděleně. Uzly, které mají tendenci být buď oba pozitivní, nebo oba negativní současně, mají silné pozitivní váhy, zatímco ty, které mají tendenci být opačné, mají silné negativní váhy. Hebbovské učení se užívá u asociativních pamětí, jako je lineární autoasociativní paměť, bidirektní heteroasociativní paměť nebo Hopfieldova síť. [25]

2.4.3.2 Přeučení

Přeučení je stav, kdy je systém příliš přizpůsoben množině trénovacích dat, ale nemá schopnost generalizace a selhává na validační množině dat. To se může stát např. při malém rozsahu trénovací množiny nebo pokud je systém příliš komplexní, např. příliš mnoho skrytých neuronů v neuronové síti. Řešením je zvětšení trénovací množiny, snížení složitosti systému nebo různé techniky regularizace, jako je zavedení náhodného šumu, zavedení omezení na parametry systému, které v důsledku snižuje složitost popisu naučené funkce, nebo předčasné ukončení učení. V průběhu učení se sledují chyby na validační množině a učení se ukončí ve chvíli, kdy se chyba na této množině dostane do svého minima. [25]

2.4.4 Hyperparametry neuronových sítí

2.4.4.1 Hyperparametry

Hyperparametry jsou parametry, jejichž hodnoty nastavuje uživatel před tréninkem modelu, např. počet vrstev neuronů. Ovlivňují chování a výkon modelu. Jsou dva typy hyperparametrů, kontinuální a diskrétní. Kontinuální nese svůj název, kvůli možnosti výběru své hodnoty z intervalu, zatímco diskrétní hyperparametry lze nastavit pouze na omezený počet možností. [29]

2.4.4.2 Parametry

Parametry jsou interní pro model. Jsou během tréninku optimalizovány modelem. Typickými parametry jsou koeficienty lineárních regresních modelů, či centroidy klastrů při klastrování. [29]

2.4.4.3 Optimalizace hyperparametrů

Hledá nejoptimálnější sadu hyperparametrů pro konkrétní neuronovou síť. Je spousta metod, které se snaží najít nejlepší hyperparametry. Těmi nejběžnějšími je vyhledávání v mřížce (grid search), náhodné vyhledávání (random search), Bayesovská optimalizace (Bayesian Optimization), Optimalizace založená na přechodech (Gradient-based Optimization) a automatické strojové učení (Automated Machine Learning). [30] [31]

2.4.4.4 Rychlost učení (learning rate)

Tento hyperparametr kontroluje jak velký krok udělá optimalizátor během každé iterace trénování. Nastavení na příliš malou rychlost může mít za následek pomalou konvergenci, avšak nastavení velké rychlosti může vést k nestabilitě a divergenci. Běžnou hodnotou je 0.01 nebo 0.001. [32]

2.4.4.5 Epochy

Tímto hyperparametrem nastavujeme počet předání celé trénovací sady při trénování modelu. Pokud počet epoch zvýšíme, může to pomoci zlepšit výkon modelu, ale riskujeme tím přeučení modelu. Běžnými hodnotami jsou 16, 32 nebo 64. [32]

2.4.4.6 Počty a typy vrstev neuronů

Počty vrstev určují hloubku modelu, pokud nastavíme moc komplexní model s mnoha vrstvami, hrozí, že se model přeučí. Tímto hyperparametrem ovlivňujeme složitost modelu a jeho možnost se učit. Běžně se nastavují 2-4 vstrvy modelu. Typy vrstev, které můžeme nastavit jsou například Dense, Conv2D či LSTM. [32]

2.4.4.7 Počty neuronů ve vrstvách

Určuje šířku modelu a jeho schopnost reprezentovat složité vlastnosti, které se v datech nachází. Vysokým počtem neuronů se zvyšuje pravděpodobnost pro přeučení modelu. Běžné nastavení může představovat 128 neuronů v první vrstvě, 32 ve druhé a 16 ve třetí vrstvě. [32]

2.4.4.8 Aktivační funkce

Tento hyperparametr ovlivňuje jak jsou výstupy vrstev transformovány. Běžně nastavenými funkcemi jsou sigmoid, tanh nebo ReLU (Rectified Linear Unit). [32]

2.4.4.9 Dropout

Tento hyperparametr náhodně vypíná neurony během tréninku a nastavuje se jako zábrana přeučení. Běžné hodnoty jsou 0, 0.2, 0.4 či 0.5. [32]

2.4.5 Předzpracování dat (data preprocessing)

Standardně se data normalizují či škálují pomocí MinMaxScaleru, či jiného nástroje pro lepší výkonnost modelu. Pokud jsou hodnoty na škále celých čísel 1-10, model by takto velké skoky v hodnotách nemusel dobře zpracovat. Pokud ale hodnoty přeškálujeme na desetinné hodnoty mezi nulou a jedničkou, skoky nebudou tak signifikantní a model se může na datech lépe naučit. Před samotným spuštěním modelu je důležité rozdělit svá data na trénovací a testovací sady. Běžně se data dělí v poměru tréninková : testovací sada 90 : 10 nebo 80 : 20. [25]

3. Řešení

3.1 Datasety

Datasety byly využity v rámci spolupráce s Ostravskou univerzitou a Univerzitou Hradec Králové. Jedná se o části datasetů z projektu „Healthy Aging in Industrial Environment (HAIE) CZ.02.1.01/0.0/0.0/16_019/0 000798“, který je spolufinancován Evropskou unií.

3.1.1 Soubor se základními informacemi

Obsahuje základní informace o uživatelích: id, stav aktivity, pohlaví, výška, váha a datum narození. V souboru se nachází informace o 497 uživatelích.

3.1.2 Soubory s aktivitou

Každý uživatel má svůj soubor s jeho zaznamenanou aktivitou po dobu jednoho roku. Soubor je tvořen maticí, ve které řádky reprezentují dny a sloupce minuty dne. Za každou minutu daného dne je zapsaná hodnota z následující škály:

- NaN – hodnota nebyla zaznamenaná
- 1 – spánek
- 2 – sedavost
- 3 – nízká aktivita
- 4 – střední aktivita
- 5 – vysoká aktivita

3.1.3 Soubor s kategorizací uživatelů

Obsahuje typově stejné informace jako první soubor: id, věk, pohlaví, výška, váha, bmi, vfat mass. Tento dataset obsahuje informace o 324 uživatelích, bylo možné pracovat pouze s 297 z důvodu duplicitních záznamů a předčasně ukončených měření. Navíc obsahuje klasifikaci uživatelů do kategorií (0 = active, 1 = inactive, 2 = obese).

3.2 Použité metody

3.2.1 K-means algoritmus

Tento algoritmus má spoustu výhod. Dá se dobře škálovat na velké množství vzorků. Garantuje konvergenci, což znamená, že po konečném počtu iterací vždy dospěje k ustálenému řešení, kde se přiřazení bodů ke klastrům a jejich centroidy již nemění, ačkoli může skončit v lokálním minimu. Má jednoduchou implementaci a rychlý výpočet pro dobře definované shluky.

Algoritmus má i své zápory. Požaduje předem specifikovat počet shluků, do kterých se mají body zařadit. Špatně reaguje na protáhlé či nepravidelné tvary. Výsledky silně závisí na inicializaci centrodů, při jejich změně to může vést k různým výsledkům. Inercie, jejíž hodnotu algoritmus optimalizuje směrem k jejímu minimu, nemusí být vždy vhodná pro všechny typy datasetů. Dále je náchylný k zachycení v lokálních minimech a citlivý na šum a odlehlé hodnoty. Vyžaduje numerické hodnoty, a tak ho nelze použít přímo na kategorie či binární data.

3.2.2 DBSCAN

Výhodou použití algoritmu DBSCAN je, že umí identifikovat klastry libovolného tvaru, na rozdíl od K-means, který předpokládá konvexní tvary. Není nutné předem specifikovat počet shluků, je nutné předem nastavit pouze parametry eps a min samples.

Na druhou stranu se algoritmus potýká i s nevýhodami. Mezi ně patří citlivost výsledků na výběr parametrů eps a min samples. U parametru eps vznikají největší potíže se správným výběrem, pokud se zvolí příliš malé, většina datasetu nebude zařazena do shluků a bude označena jako šum. Při zvolení naopak příliš velkého eps dochází ke slučování blízkých klastrů, případně k vytvoření jednoho velkého shluku pro celý dataset. Je nevhodný pro velmi velké datové soubory, kvůli jeho výpočetní náročnosti.

3.2.3 Agglomerative algoritmus

Tento algoritmus poskytuje hierarchii shluků, která umožňuje analyzovat data na různých úrovních detailu. Není třeba předem specifikovat počet klastrů, protože hierarchický strom (dendrogram) lze prořezat na libovolné úrovni, což umožňuje dodatečné rozhodování o počtu shluků. Další výhodou je schopnost pracovat i s nekonvexními tvary shluků.

Metoda je výpočetně náročná – její časová složitost je $O(n^3)$ a paměťová složitost $O(n^2)$, což ji činí nevhodnou pro velmi velké datasety. Odlehlé hodnoty mohou negativně ovlivnit výsledky, zejména při použití metrik citlivých na vzdálenost. Pro dosažení kvalitních výsledků je také důležité mít dobře škálovaná data, aby různé rozsahy hodnot v datasetu neovlivnily analýzu.

3.3 Metriky pro optimalizaci počtu klastrů

3.3.1 Silhouette skóre

Metrika hodnotí kohezi, tedy jak blízko jsou body k ostatním bodům ve stejném shluku, což odráží vnitřní soudržnost. Dále hodnotí separaci, tedy jak daleko jsou body od bodů v jiných klastrech, což vyjadřuje mezishlukovou separaci.

Pokud je výsledkem skóre blízké 1, znamená to, že body jsou dobře přiřazeny ke svým klastrům. Naopak skóre blízké -1 naznačuje, že body jsou přiřazeny špatně. Pokud je skóre rovno 0, znamená to, že body se nacházejí na hranici mezi dvěma shluky.

3.3.2 Davies-Bouldin skóre

Hodnotí kompatibilitu shluků, tedy jak dobře jsou klastry oddělené. Při hodnocení se zohledňuje velikost a variabilita jednotlivých klastrů.

Nižší hodnoty skóre znamenají lepší klastrování, protože naznačují, že shluky jsou dobře oddělené a mají malou variabilitu uvnitř.

3.3.3 Calinski-Harabasz skóre

Skóre hodnotí mezishlukovou separaci a vnitřní kohezi, tedy jak homogenní jednotlivé shluky jsou. Tento ukazatel pomáhá posoudit kvalitu shlukování na základě rozdílů mezi klastry a jejich vnitřní soudržností.

Výsledky vyšších hodnot indikují lepší klastrování, protože naznačují, že klastry jsou dobře oddělené a zároveň kompaktní.

3.3.4 PCA (Principal Component Analysis)

PCA je metoda, která efektivně snižuje počet dimenzí dat, čímž zlepšuje výpočetní náročnost a eliminuje redundantní informace. Klíčovou výhodou je, že maximální možná část variability dat je zachována v prvních několika hlavních komponentách. Díky tomu lze PCA využít i k vizualizaci vysokodimenzionálních dat ve 2D nebo 3D prostoru, což usnadňuje jejich analýzu.

Na druhou stranu snížení dimenzí může vést ke ztrátě některých informací, zejména pokud je použito příliš málo komponent. Další nevýhodou je výpočetní náročnost. Výpočet vlastních čísel a vlastních vektorů velkých matic může být časově náročný, zejména pro velmi rozsáhlé datasety.

3.4 Použití metod

3.4.1 Klastrová analýza

Nejprve bylo provedeno předzpracování dat. Klastrovou analýzu jsem dělala na jednotlivých uživatelích a při pouhém nahrazení NaN hodnot nulou byla vždy část shluků nepoužitelná (u K-means a Agglomerative algoritmu při nastavení 11 klastrů, které vyšlo pro oba algoritmy u vybraného uživatele jako optimální dle výše uvedených metrik, jsou nepoužitelné 4 shluky), a tak jsem zvolila přístup odstranění řádků, které obsahují NaN hodnotu. V tomto případě nastal problém s nedostatkem dat. V nejlepším případě zbyla jednomu uživateli 1/4 řádků (přibližně 80 z 350), a tak jsem zmírnila podmínky pro odstranění řádků a odstranila jsem dny, kde byla zapsána více než 5krát NaN hodnota. Za této podmínky zůstanou v nejlepším případě 2/3 řádků jednoho uživatele (přibližně 250 z 350), kde již bude možné najít trendy ve všech řádcích uživatele a tím pádem i ve všech klastrech, do kterých tyto řádky jednotlivé algoritmy zařadí.

Dále bylo důležité vybrat počet shluků u K-means a Agglomerative algoritmů. Byly použity 3 metriky (Silhouette skóre, Davies-Bouldin skóre a Calinski-Harabasz) a v nich jsem hledala optimální počet klastrů mezi čísly 10 – 15. Rozhodla jsem se tak, ačkoli metriky
preferovaly nižší počet, protože při vyšším počtu je možné sledovat podrobné vzorce chování uživatele.

Pro výběr optimálnějších hodnot u parametrů min samples a eps probíhal iterativní experimentací s cílem najít kombinaci hodnot, u kterých algoritmus rozdělí data do více než jednoho klastru.

3.4.2 Neuronové sítě

3.4.2.1 Dopad velikosti trénovací množiny na přesnost modelu

Pro tento problém se snižoval počet uživatelů zastoupených v každé kategorii. Model klasifikoval každý den uživatelů do kategorie (aktivní, neaktivní, obézní). Nejdříve byli vybráni náhodní uživatelé z každé kategorie, dle specifikovaného počtu, a poté se jejich data dělila na trénovací a validační sadu v poměru 90 : 10.

Architektura modelu:

```

model = Sequential([
    Input(shape=(X_train.shape[1], 1)),
    Conv1D(8, 2, activation='relu'),
    MaxPooling1D(2),
    Conv1D(16, 2, activation='relu'),
    MaxPooling1D(2),
    Conv1D(16, 2, activation='relu'),
    MaxPooling1D(2),
```

```
    Conv1D(32, 2, activation='relu'),
    MaxPooling1D(2),
    Conv1D(32, 2, activation='relu'),
    MaxPooling1D(2),
    Flatten(),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax')
])

model.compile(Adam(0.004), 'sparse_categorical_crossentropy', metrics=['
    ↪ accuracy'])
early_stop = EarlyStopping(monitor='val_loss', patience=3,
    ↪ restore_best_weights=True)

model.fit(
    X_train, y_train,
    epochs=30,
    batch_size=64,
    validation_split=0.2,
    callbacks=[early_stop]
)
```

Ukázka kódu 3.1 Architektura modelu

3.4.2.2 Klasifikace uživatelů

Pro problém klasifikování uživatelů do kategorií byla neuronová síť natrénována na datech klasifikovaných uživatelů. Po tréninku model obdržel data pouze jednoho uživatele a výsledkem bylo kategorizování jednotlivých dnů (řádků) a poté označení uživatele do kategorie s největším zastoupením.

Model byl natrénován na náhodně vybraných 60 lidech z každé kategorie a byla vybrána stejná architektura modelu jako pro předchozí problém.

4. Výsledky

4.1 Klastrová analýza

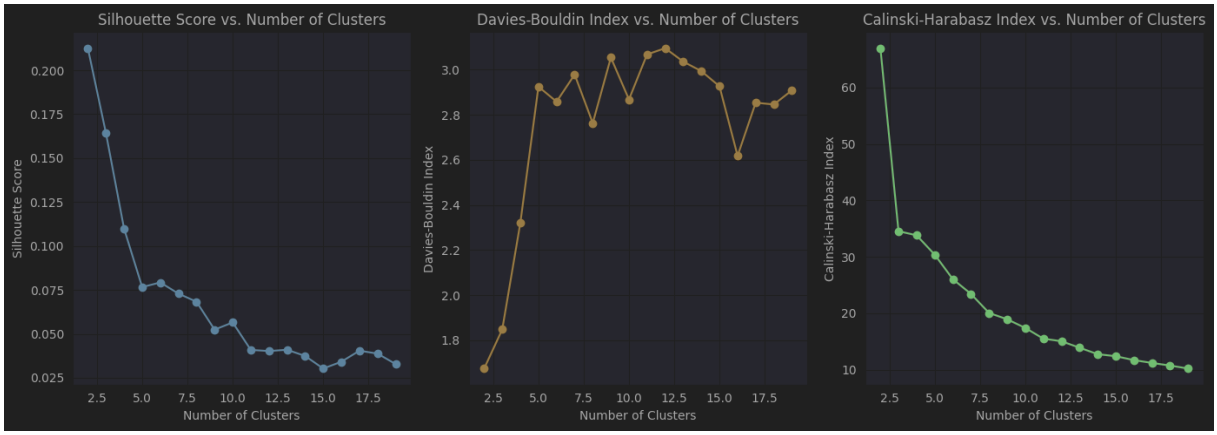
4.1.1 Výběr optimálního počtu klastrů u K-means algoritmu

V rámci analýzy klastrování metodou K-means pro uživatele s identifikátorem 1289 byly použity tři validační metriky k určení optimálního počtu klastrů v rozsahu 2–20. Výsledky ukazují odlišné preference jednotlivých metrik, což vyžadovalo pečlivé vyhodnocení.

Hodnoty Silhouette Score zůstaly celkově nízké, avšak s relativní stabilitou v oblasti 10–20 shluků. V tomto rozsahu nebyly zaznamenány výrazné změny, což naznačuje omezenou schopnost algoritmu výrazně zlepšit separaci datových bodů při vyšším počtu klastrů. Mírné zlepšení metriky bylo pozorováno v intervalu 12–14 shluků, což naznačuje potenciálně lepší vnitroshlukovou homogenitu v tomto pásmu.

Davies-Bouldin Index, který preferuje nižší hodnoty jako indikátor kvality, identifikoval jako optimální počet 16 klastrů. Tento výsledek odráží nejlepší poměr mezi vnitřní podobou klastrů a jejich vzájemnou odlišností podle této metriky. Naproti tomu Calinski-Harabasz Index vykazoval klesající trend s rostoucím počtem klastrů, přičemž vyšší relativní hodnoty v rozmezí 12–14 shluků signalizovaly lepší poměr mezi mezishlukovou rozptýleností a vnitřní kompaktností.

Závěrečné rozhodnutí o volbě 13 shluků vychází ze syntézy všech tří metrik. Ačkoli Davies-Bouldin Index upřednostňoval vyšší počet klastrů, kombinace stabilního Silhouette Score a lokálního maxima Calinski-Harabasz Indexu v oblasti 12–14 vedla k volbě kompromisního řešení. Tento přístup umožňuje zachovat dostatečnou granularitu klastrování při respektování trendů ve vývoji validačních kritérií.

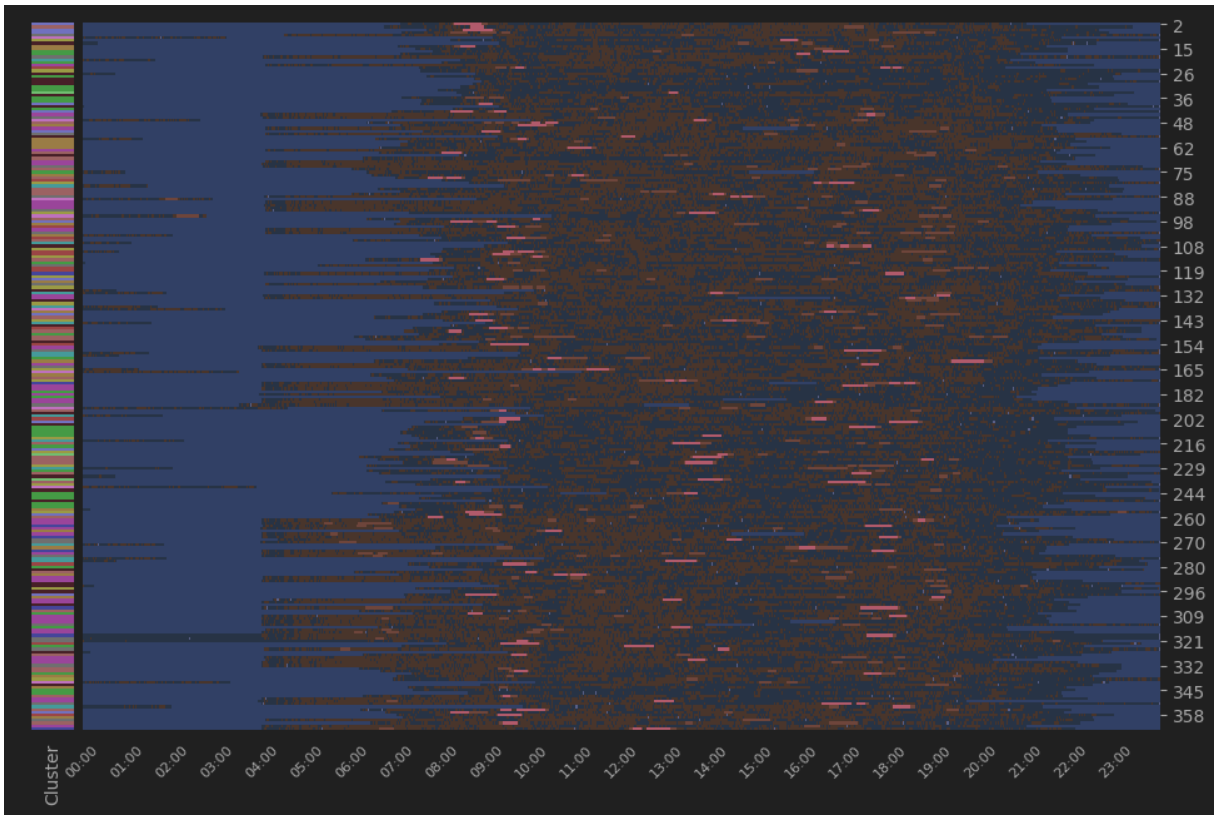


Obrázek 4.1: Výsledky metrik pro hledání optimálního počtu klastrů u K-means algoritmu.

4.1.2 Analýza časového rozložení klastrů K-means algoritmu pomocí heatmapy

Heatmapa s vertikální barevnou legendou při levém okraji vizualizuje rozložení shluků v časové ose podle příslušnosti jednotlivých dní. Primárním cílem této analýzy je detekovat rozsáhlé bloky po sobě jdoucích dní přiřazených ke stejnému shluku. Tyto bloky by mohly naznačovat výkyvy od standardního chování v daném časovém období.

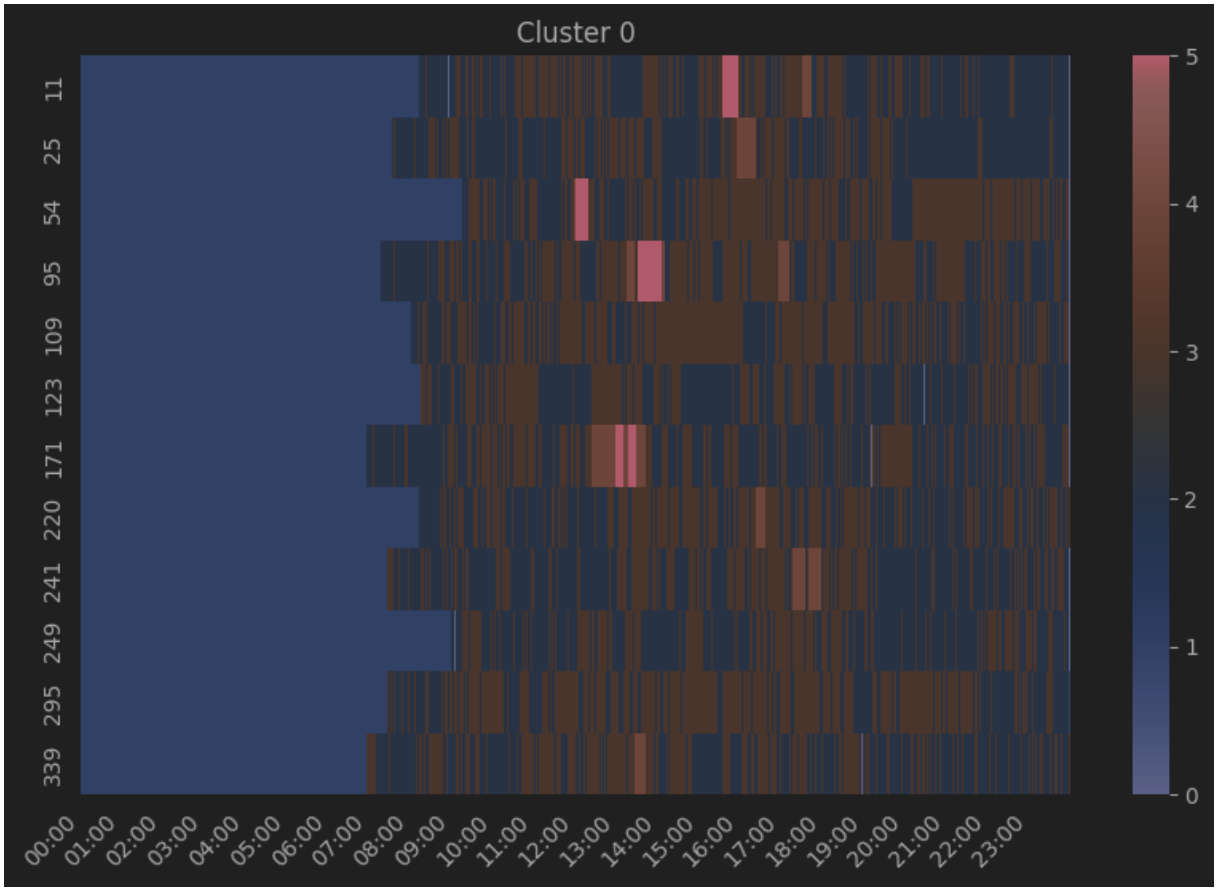
Výsledná vizualizace však tuto situaci nepotvrzuje. Jednotlivé klastry jsou v časové řadě nespojitě rozptýleny bez vytvoření výrazných souvislých sekvencí. Dny spadající do stejných skupin pocházejí z různých období měření, což naznačuje, že identifikované vzorce chování nejsou vázány na konkrétní časové úseky.



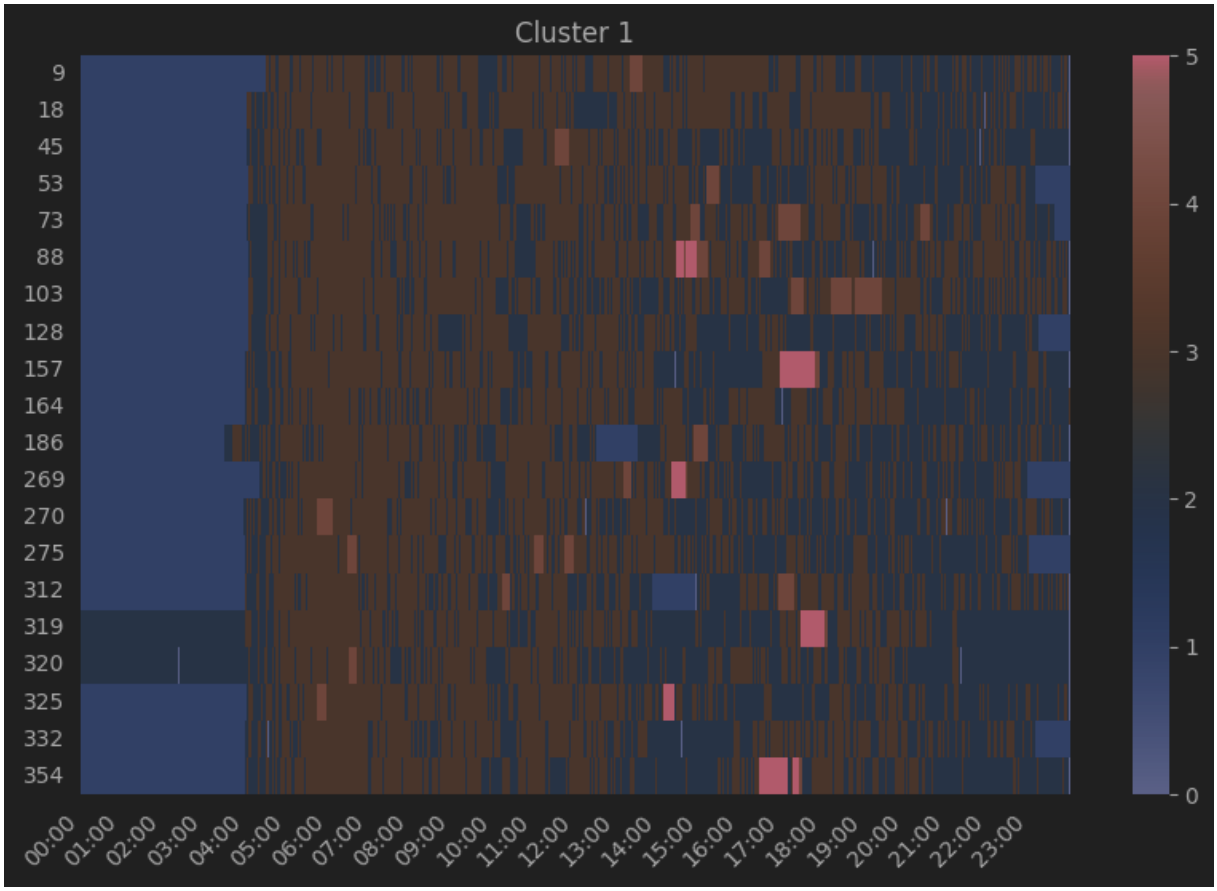
Obrázek 4.2: Heatmapa s rozložením klastrů K-means algoritmu

4.1.3 Zobrazení jednotlivých klastrů K-means algoritmu

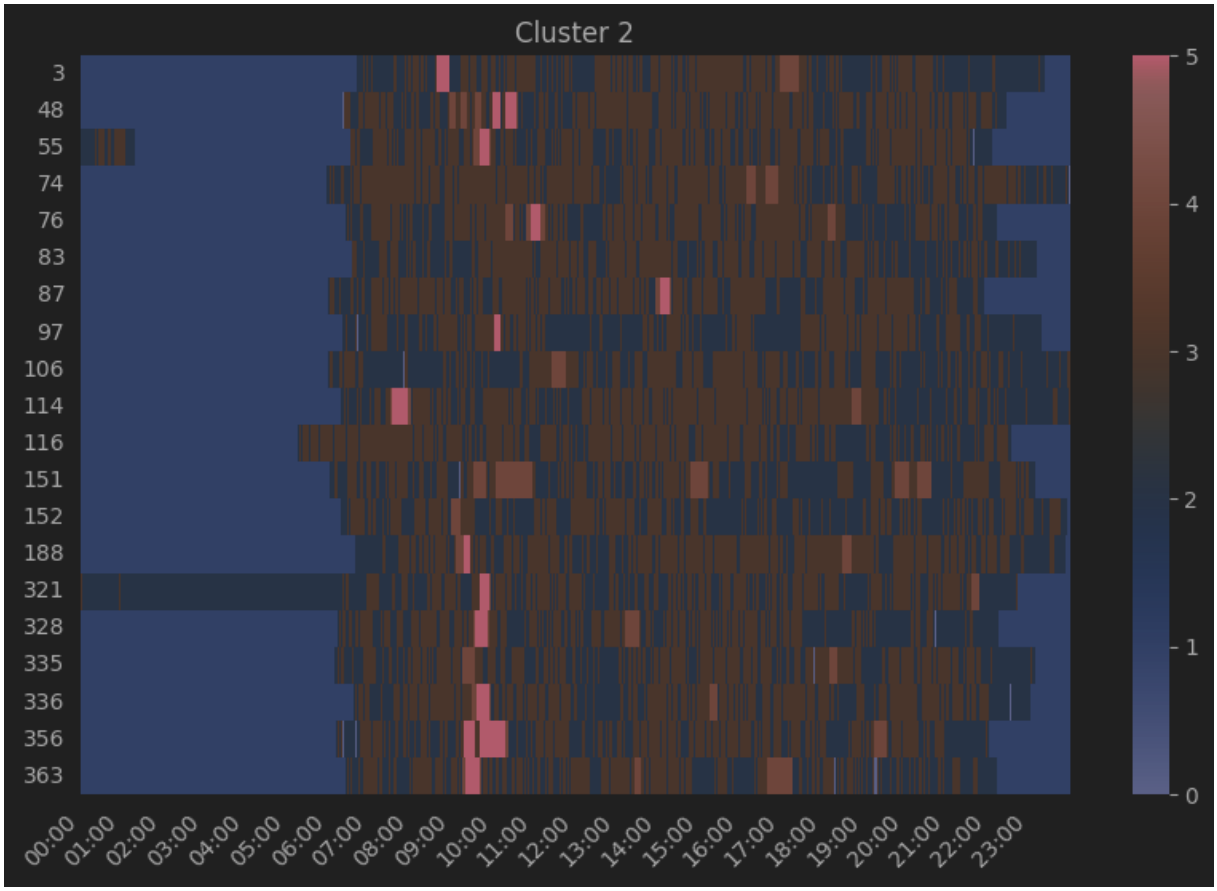
Číslování klastrů není určeno na základě jakýchkoli faktorů, je zcela náhodné, nenaznačuje žádné vlastnosti klastrů ani dat v nich.



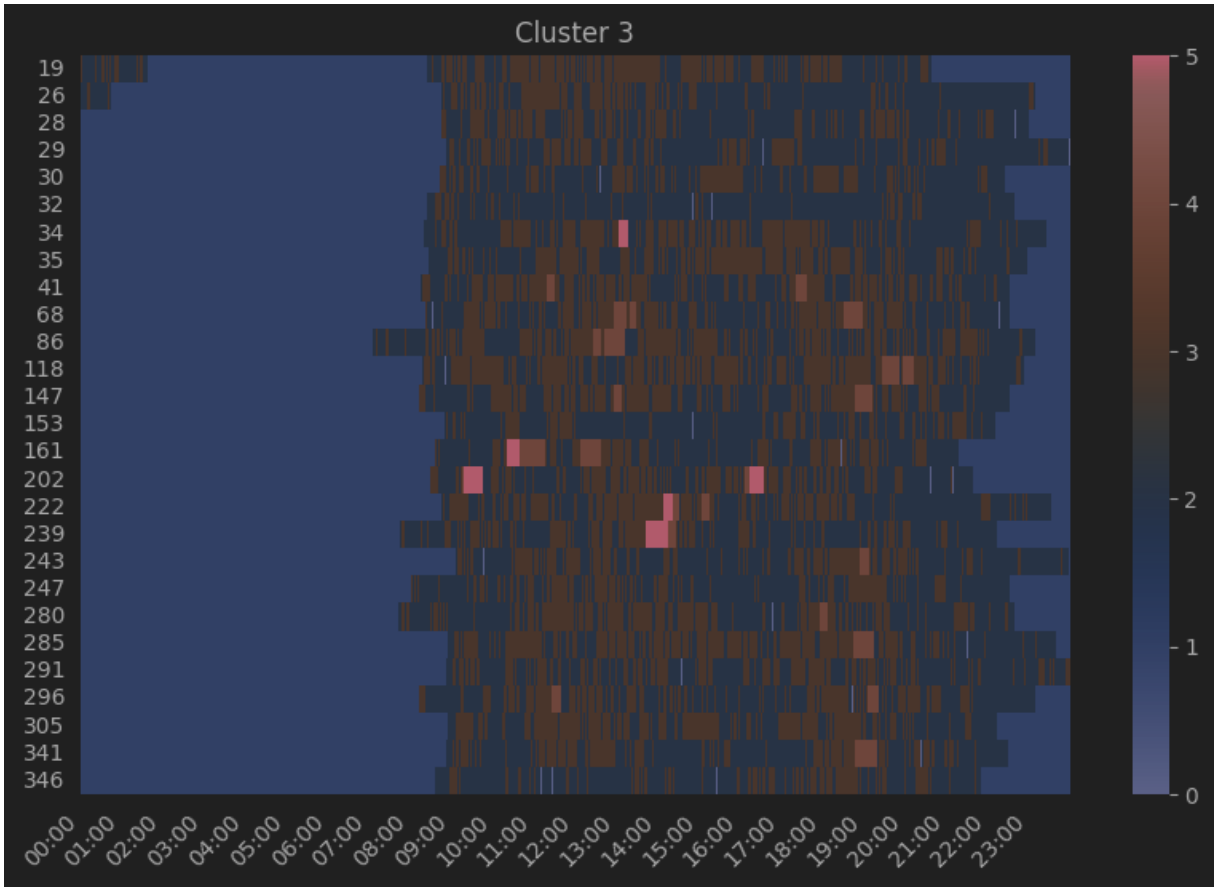
Obrázek 4.3: Shluk 0 zahrnuje dny, kdy uživatel obvykle vstával kolem 8. hodiny ranní. Po zbytek dne střídal sedavé aktivity s převážně nízkou aktivitou, doplněné občasnými epizodami střední a vysoké aktivity. Tyto vyšší úrovně aktivity se nejčastěji objevovaly v odpoledních hodinách, zejména mezi 14. a 17. hodinou.



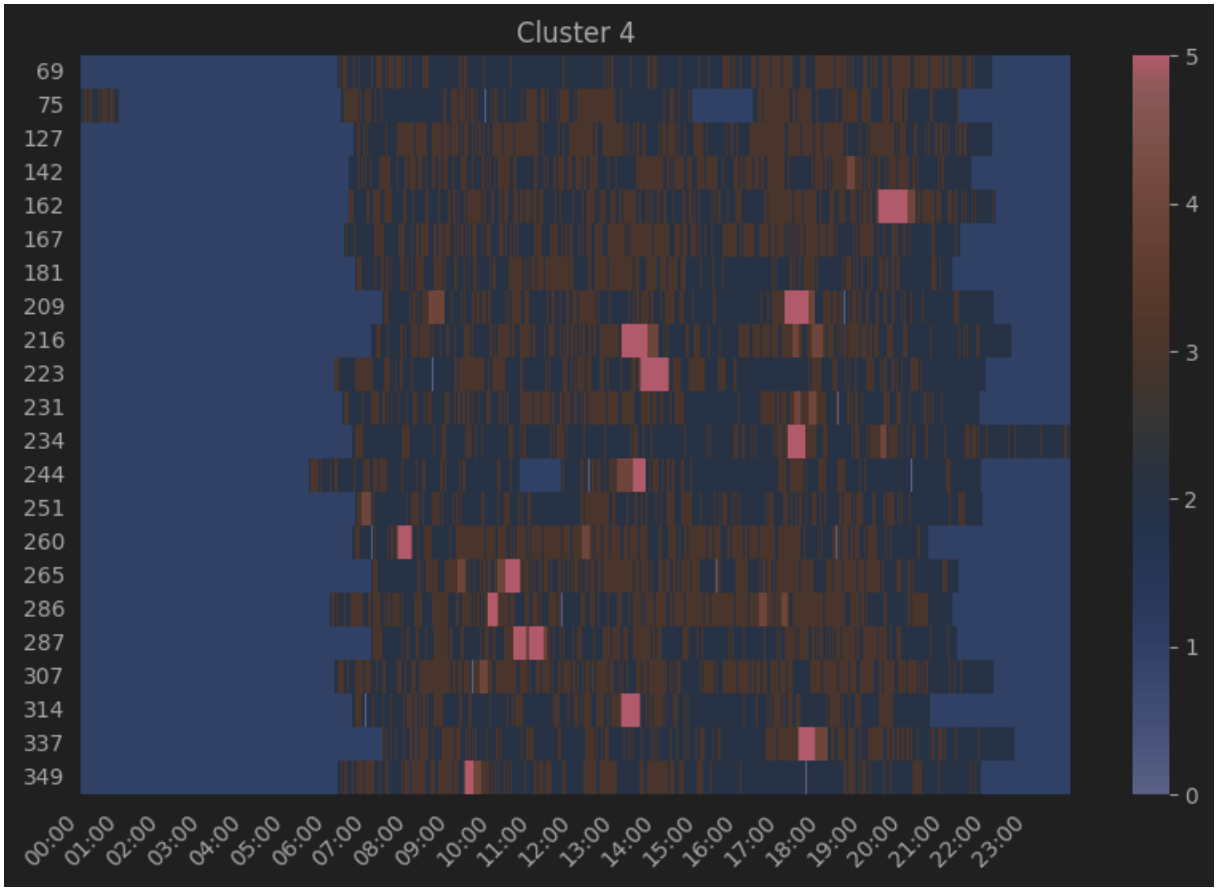
Obrázek 4.4: V klastru 1 jsou zahrnuty dny, kdy uživatel obvykle vstával velmi brzy, kolem 4:30 ráno. Dopolední hodiny byly charakteristické převážně mírnou aktivitou, která byla často přerušována sedavými činnostmi. Na rozdíl od dopoledne se odpolední hodiny vyznačovaly výraznějším podílem sedavosti, která převažovala nad ostatními typy aktivit.



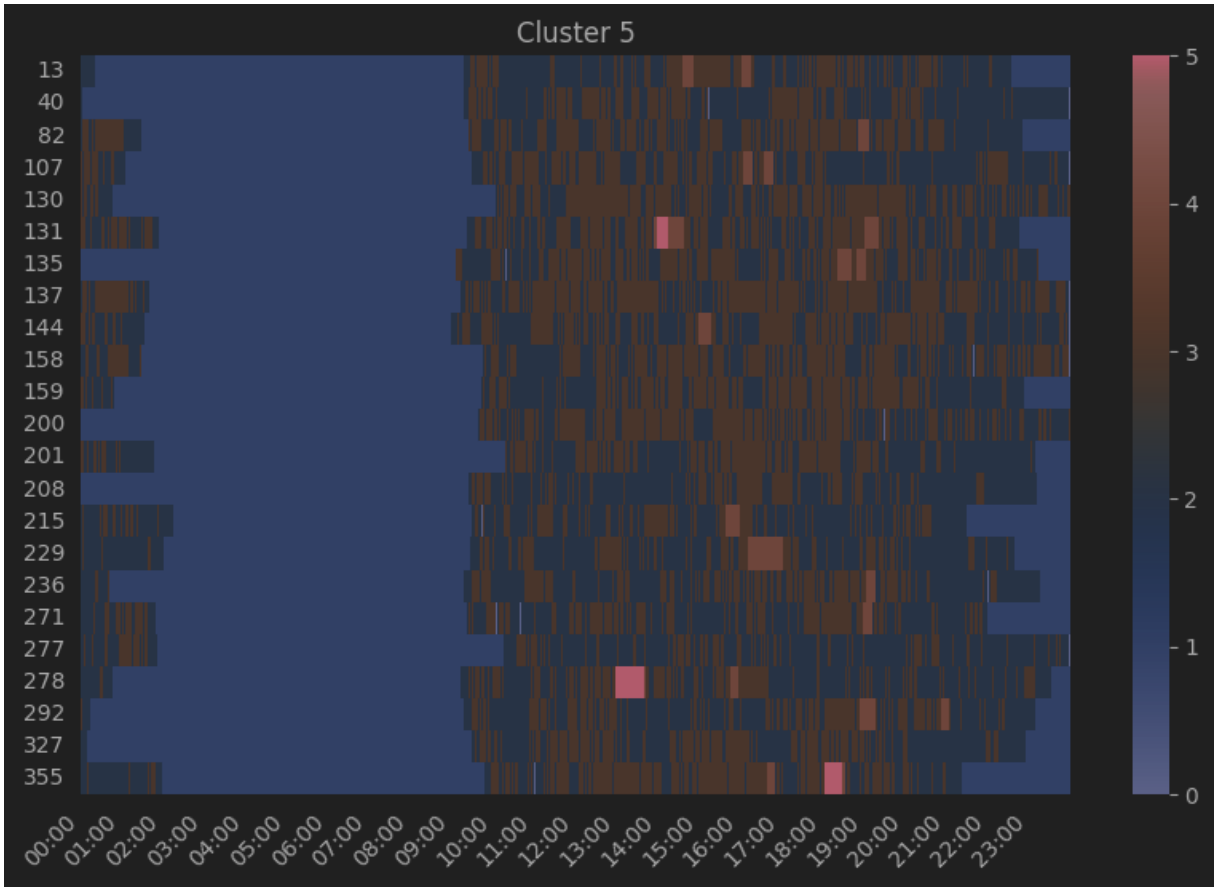
Obrázek 4.5: V klastru 2 se nacházejí dny, kdy uživatel obvykle začínal svůj den kolem 7. hodiny ranní. Po většinu dne dominovala mírná aktivita, která se pravidelně střídala se sedavostí. Občas byly zaznamenány také krátké epizody střední nebo vysoké aktivity, které však byly spíše výjimečné.



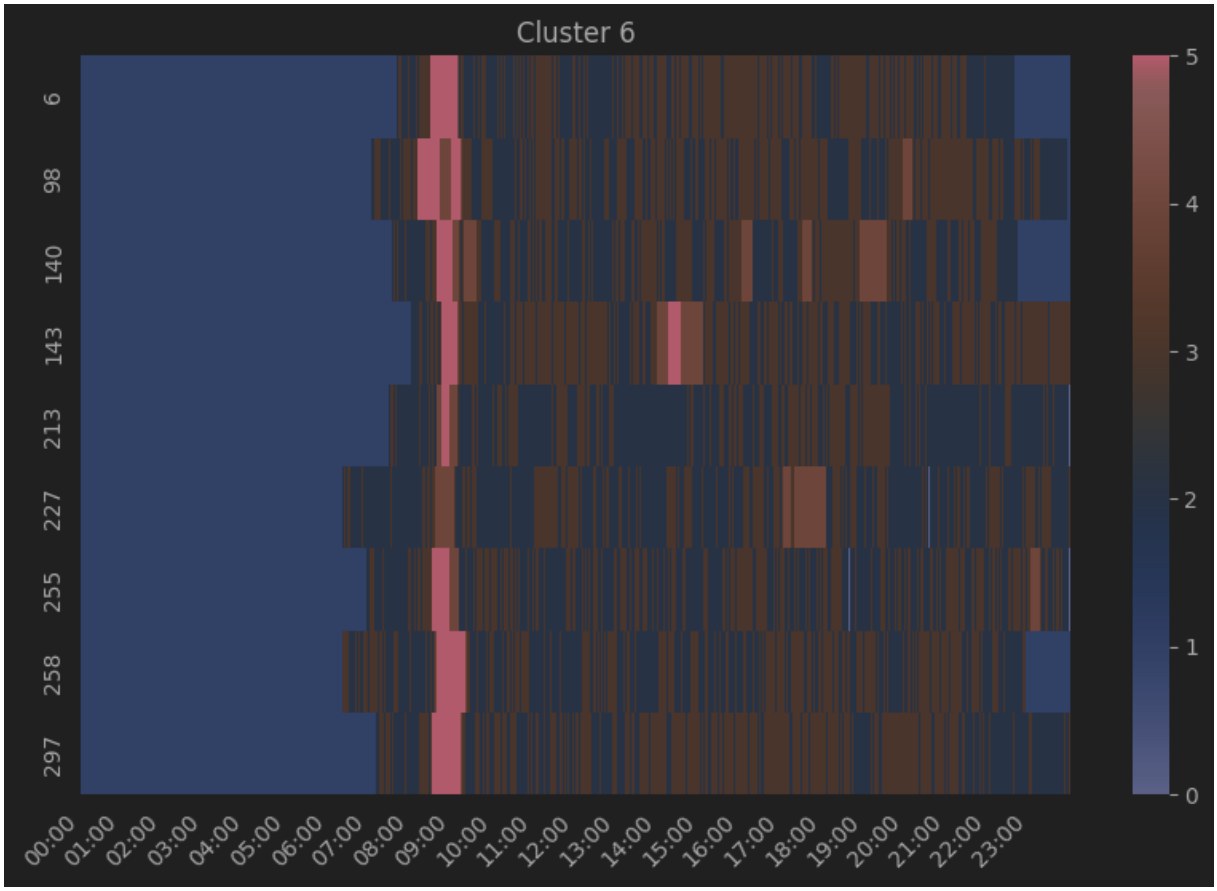
Obrázek 4.6: Shluk 3 zahrnuje dny, kdy uživatel vstával až kolem 9. hodiny ranní, což je patrné z převládající hodnoty 1, jež označuje spánek, v časných ranních hodinách. V průběhu dne dominuje sedavost, přičemž jen výjimečně byly zaznamenány krátké úseky nízké nebo střední aktivity, zejména odpoledne a podvečer. Hodnoty vysoké aktivity se v tomto klastru téměř nevyskytují.



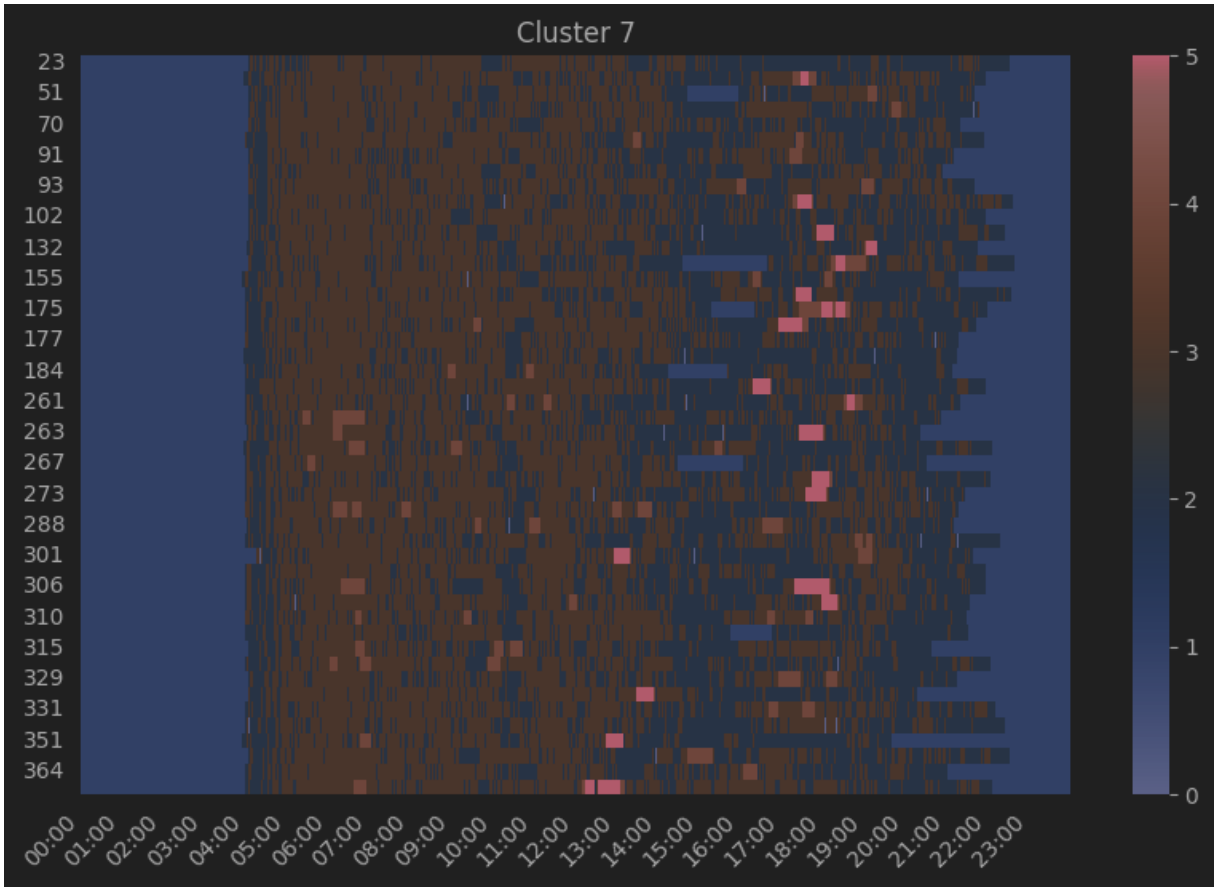
Obrázek 4.7: Shluk 4 zahrnuje dny, kdy uživatel vstával kolem 7. hodiny ranní, což je vidět na přechodu od hodnoty 1, spánku, k vyšším hodnotám. Během dne převládá kombinace sedavosti a nízké aktivity, s občasnými úseky střední aktivity, zejména v odpoledních a večerních hodinách. Hodnoty vysoké aktivity se vyskytují jen výjimečně.



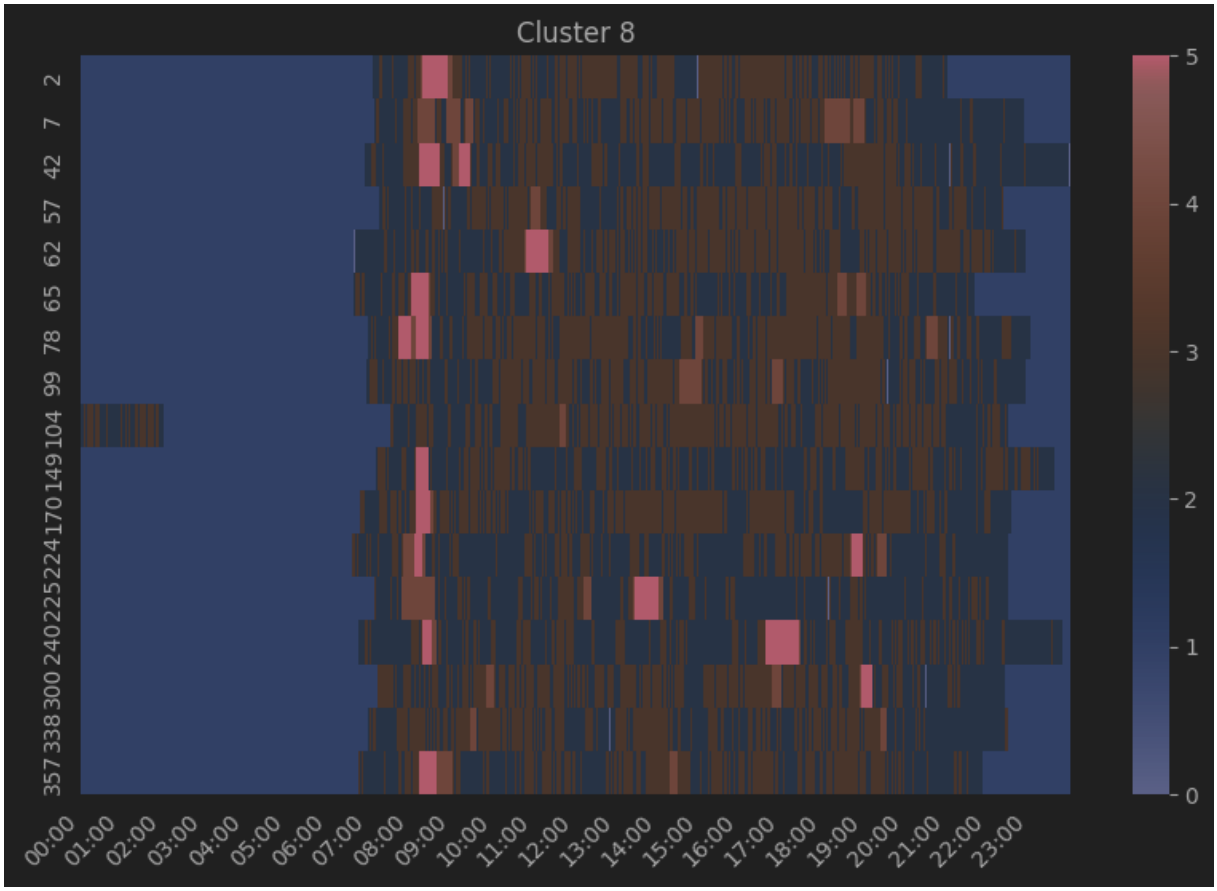
Obrázek 4.8: Do klastru 5 spadají dny, kdy byl uživatel aktivní i v pozdních nočních hodinách, konkrétně až do 2. hodiny ranní. Následně vstával kolem 10. hodiny dopoledne. Během dne jeho aktivita převážně odpovídala sedavému režimu, s občasnými úseky mírné fyzické aktivity.



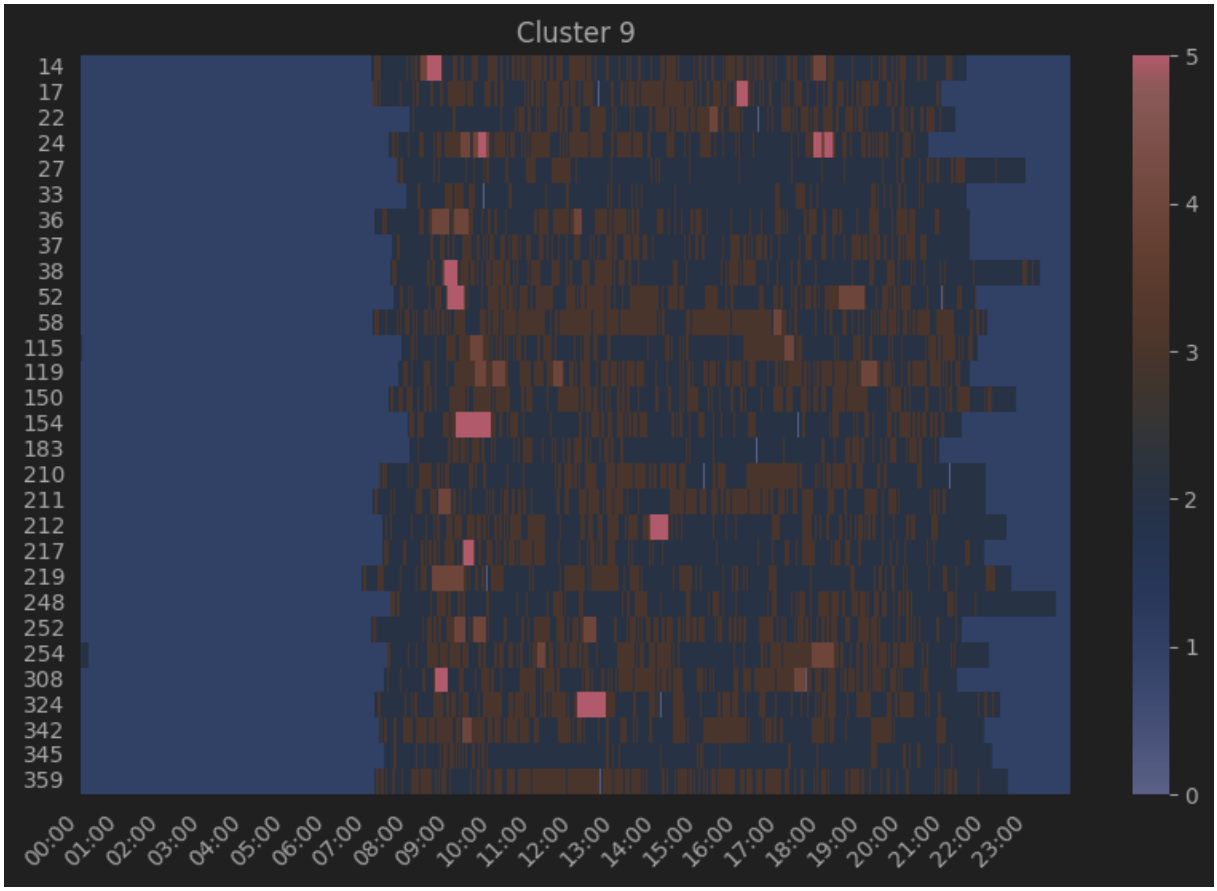
Obrázek 4.9: Shluk 6 charakterizují dny, kdy uživatel začínal den v 8 hodin ráno a následně v 9 hodin ráno vykazoval přibližně hodinovou střední až vysokou aktivitu. Během zbytku dne převládala mírná aktivita nad sedavým chováním.



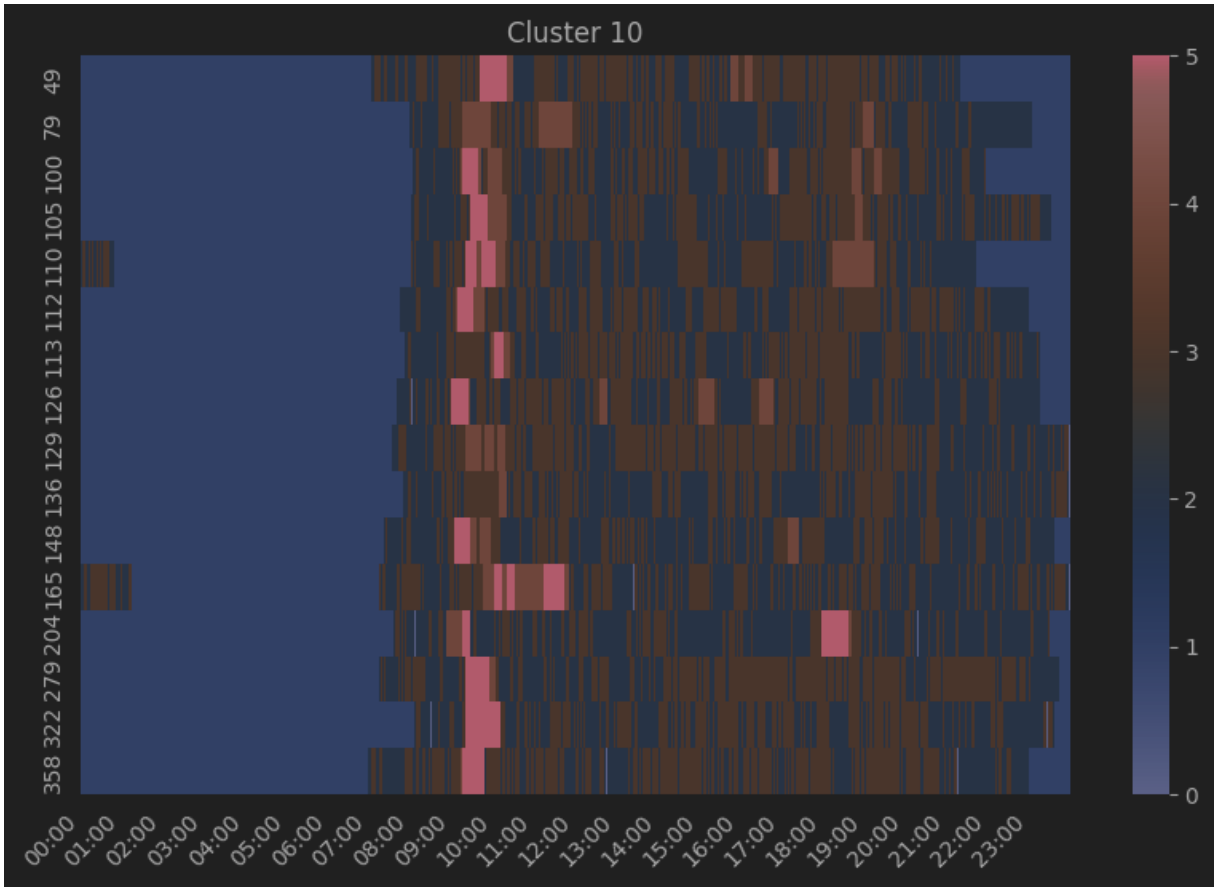
Obrázek 4.10: Shluk 7 zahrnuje dny, kdy uživatel vstával brzy ráno, přibližně ve 4:30, a během celého dopoledne vykazoval převážně mírnou aktivitu. V odpoledních hodinách se začaly objevovat hodnoty sedavého chování, které převažovaly, avšak byly příležitostně přerušovány krátkými obdobími vysoké aktivity.



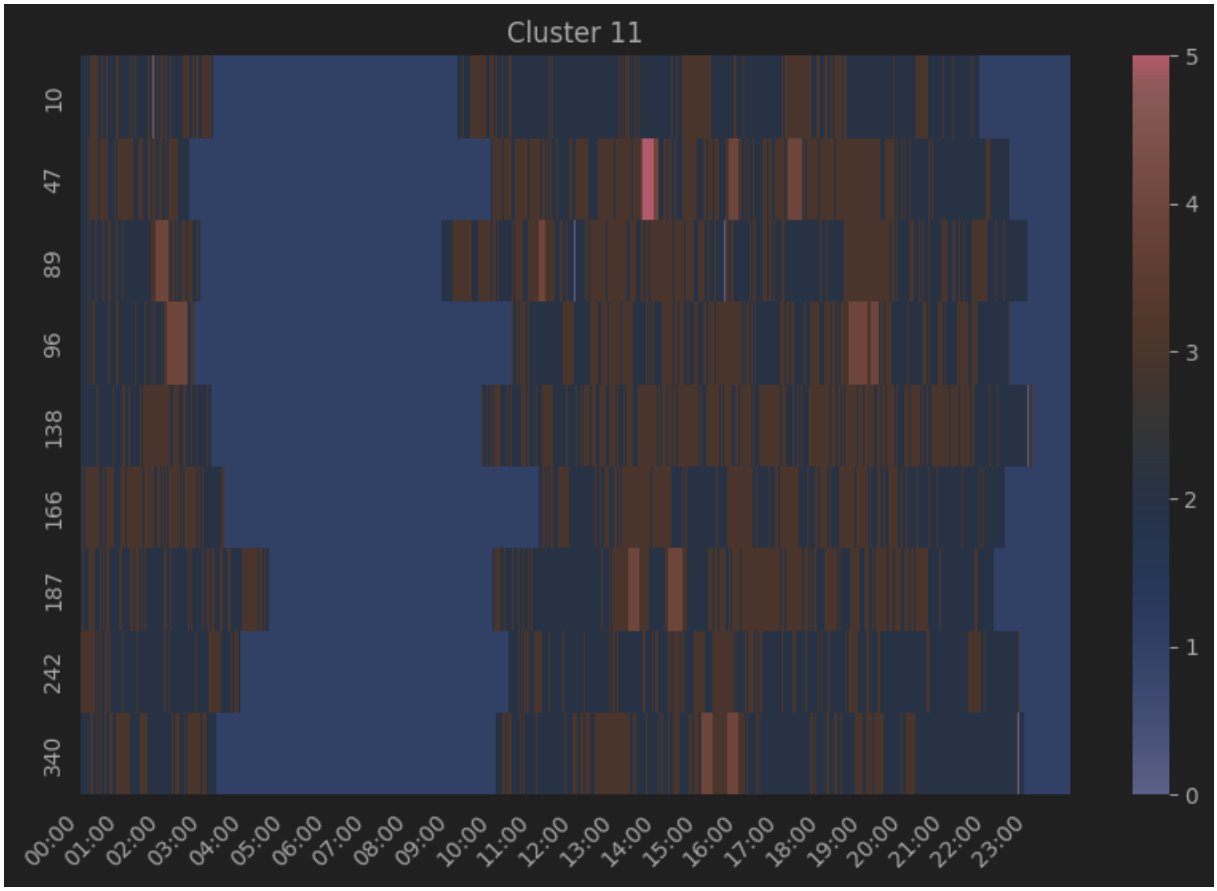
Obrázek 4.11: Klastr 8 zahrnuje dny, kdy uživatel vstával kolem 7. hodiny ráno. V některých z těchto dnů vykazoval krátce po probuzení vysokou úroveň aktivity. Večer uživatel obvykle chodil spát poměrně brzy, přibližně kolem 23. hodiny.



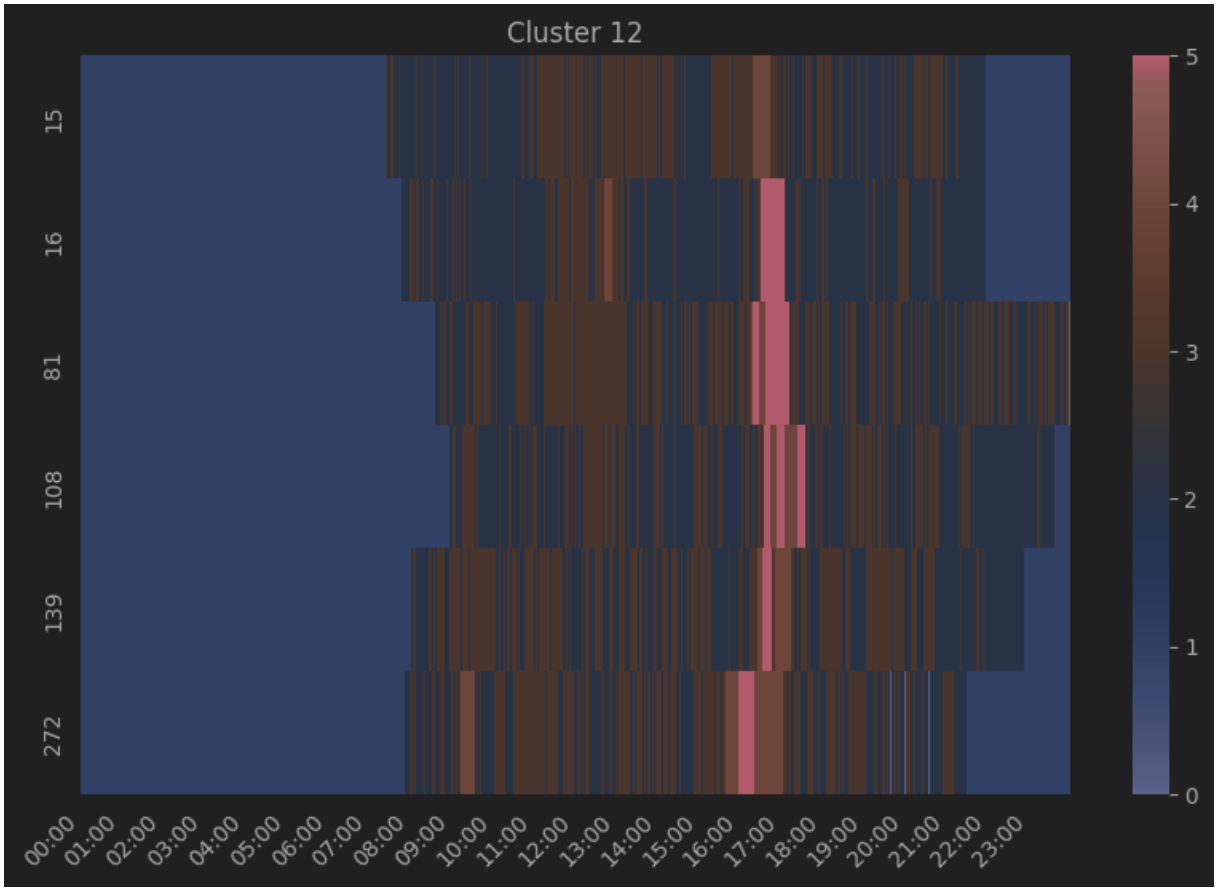
Obrázek 4.12: Klastr 9 zahrnuje dny, kdy uživatel obvykle vstával kolem 8. hodiny ráno. Během dne převládalo sedavé chování a mírná aktivita, avšak v některých dnech byla zaznamenána vysoká aktivita, která se objevovala především během několika hodin po probuzení.



Obrázek 4.13: Shluk 10 zahrnuje dny, kdy byl uživatel velmi aktivní již během první hodiny po probuzení. Úroveň aktivity dosahovala až vysokých hodnot. Po zbytek dne se střídala mírná aktivita se sedavým chováním, což naznačuje dynamický začátek dne následovaný klidnějším průběhem.



Obrázek 4.14: Klastr 11 zahrnuje dny, kdy byl uživatel aktivní v brzkých ranních hodinách, konkrétně mezi půlnocí a 3. hodinou. Následně si šel odpočinout a kolem 10. hodiny ráno vstal. Zbytek dne byl charakterizován převážně sedavým chováním a mírnou aktivitou. Večer uživatel obvykle chodil spát kolem 23. hodiny, což naznačuje pravidelný režim s noční aktivitou a klidnějším průběhem dne.

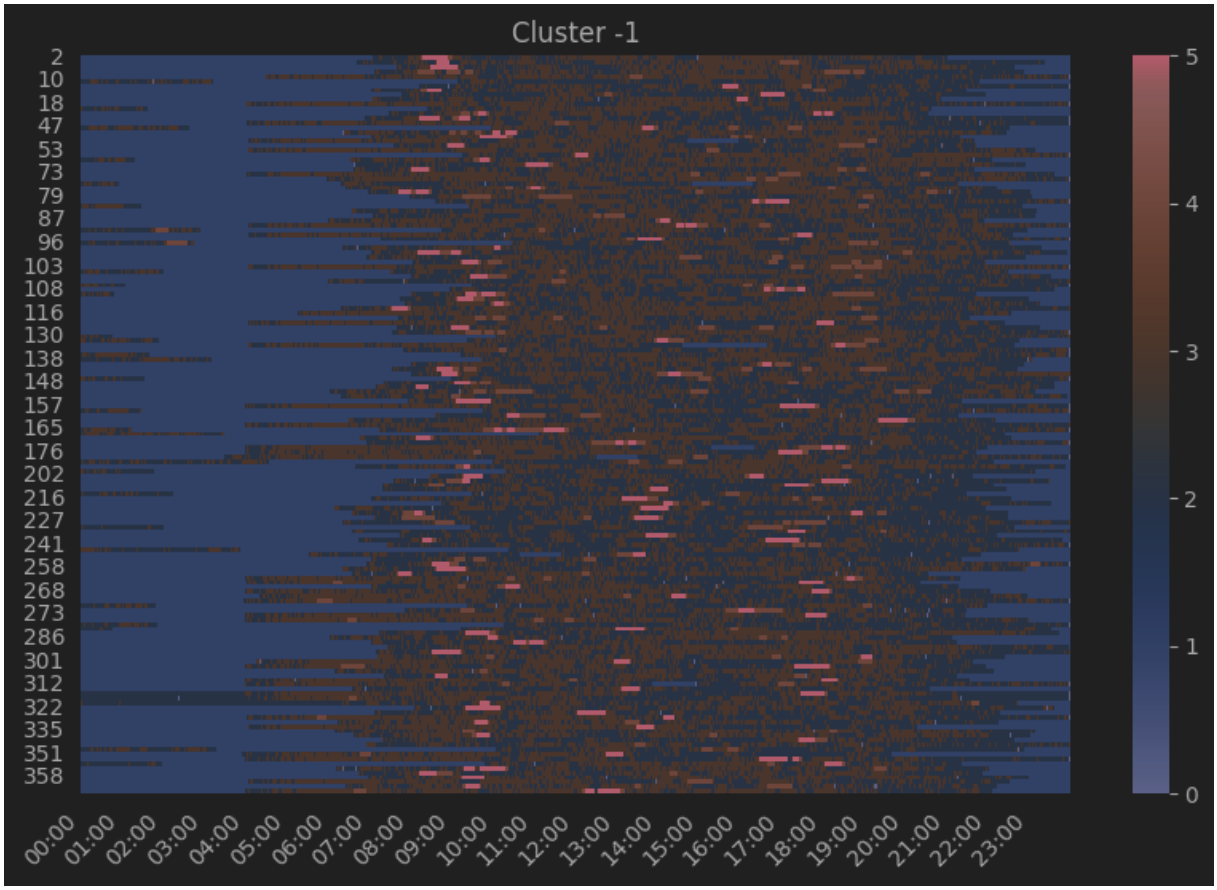


Obrázek 4.15: Shluk 12 zahrnuje dny, kdy byl uživatel vzhůru od 8. hodiny ranní. Kolem 17. hodiny odpoledne vykazoval vysokou úroveň aktivity, jež se výrazně odlišovala od zbytku dne. Svůj den obvykle zakončoval spánkem již kolem 22. hodiny večer.

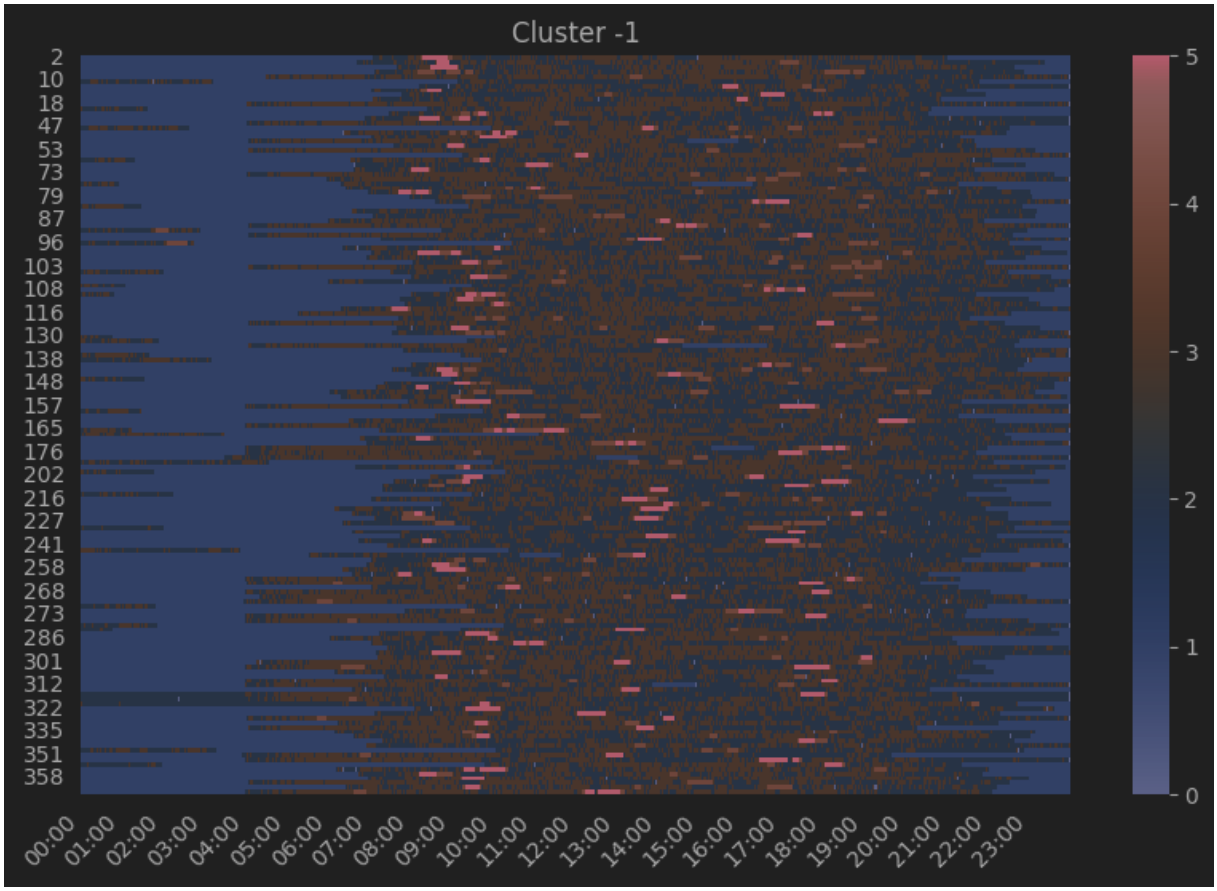
4.1.4 Vybrané parametry pro DBSCAN algoritmus

Pro analýzu dat uživatele s ID 1289 byl použit algoritmus DBSCAN s parametry eps = 22 a min samples = 6. U tohoto algoritmu není nutné sledovat, zda se shluky skládají pouze z několika po sobě jdoucích dní.

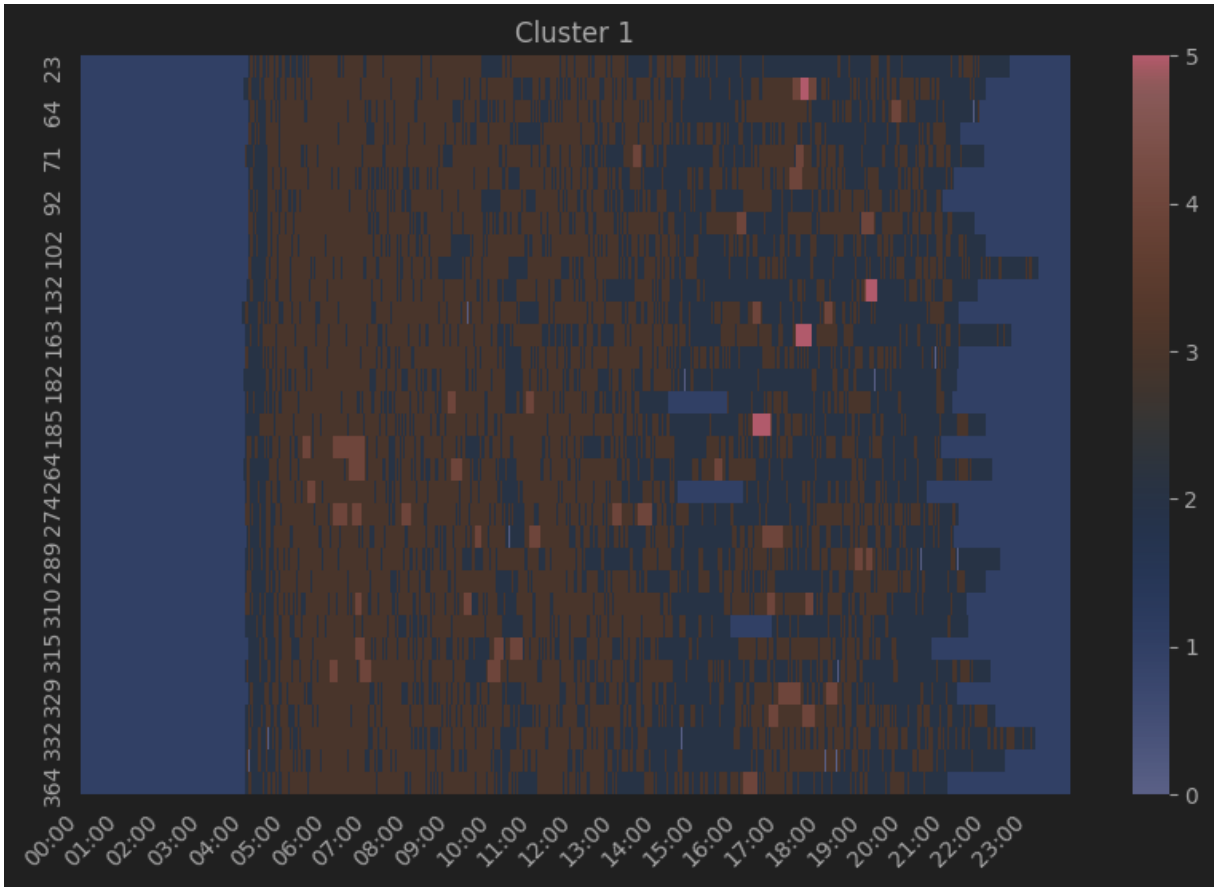
4.1.5 Zobrazení jednotlivých klastrů algoritmu DBSCAN



Obrázek 4.16: Klastr s číslem -1 je obvykle označován jako klastr s šumem (outliery), protože nezachycuje čisté nebo konzistentní vzorce chování. Nicméně, v rámci tohoto shluku lze rozlišit dva typy trendů: jeden, kde uživatel zůstával vzhůru dlouho do noci, a druhý, kde vstával velmi brzy ráno. Tyto rozdílné vzorce naznačují, že data v tomto shluku jsou rozmanitější a méně strukturovaná než v ostatních shlucích.



Obrázek 4.17: Shluk 0 zahrnuje dny, kdy uživatel nebyl dlouho do noci vzhůru a svůj den obvykle začínal mezi 8. a 10. hodinou ranní. Tento shluk tak zachycuje dny s relativně pozdním ranním režimem a absencí pozdních nočních aktivit.



Obrázek 4.18: Klastr 1 vykazuje velmi specifické chování uživatele, charakterizované tím, že uživatel vstával ve velmi brzkých ranních hodinách a to vždy ve stejný čas. Tento klastr tak zachycuje dny s vysoce pravidelným ranním režimem.

4.1.6 Výběr optimálního počtu klastrů u Agglomerative algoritmu

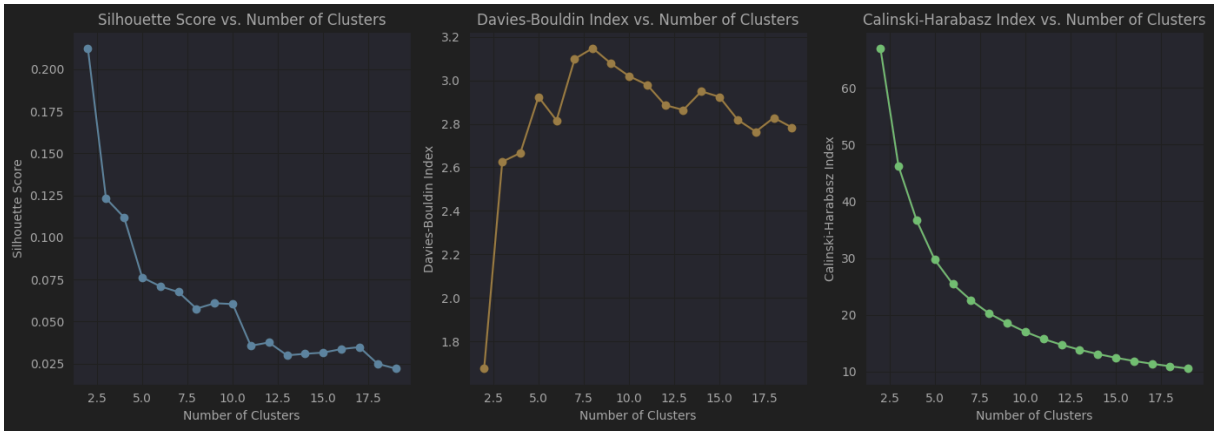
V rámci analýzy agglomerativního klastrování pro uživatele s ID 1289 byly použity tři validační metriky k určení optimálního počtu shluků v rozsahu 2–20. Výsledky ukazují odlišné preference jednotlivých metrik.

Hodnoty Silhouette Score zůstaly celkově nízké, zejména mezi 10 a 15 shluky. Nicméně, nejvyšší relativní hodnota byla pozorována u 10 klastrů. Poté skóre klesá a zůstává nízké, což naznačuje, že 10 klastrů je relativně dobrá volba z hlediska vnitřní soudržnosti shluků.

Davies-Bouldin Index, který preferuje nižší hodnoty jako indikátor kvality, identifikoval jako optimální počet klastrů 11. Poté index osciluje, avšak žádná z následujících hodnot nebyla výrazně lepší než tato. To naznačuje, že rozdělení na 11 shluků je relativně dobré z hlediska separace shluků.

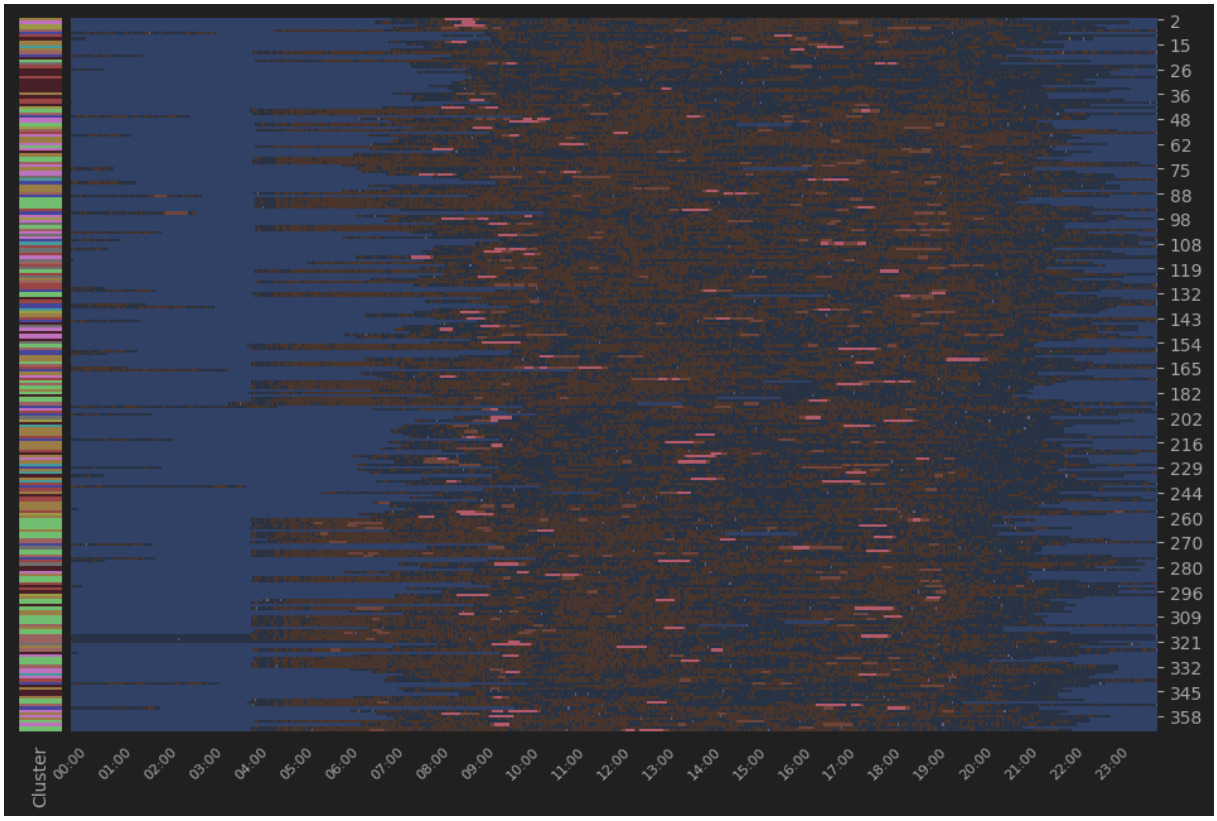
Calinski-Harabasz Index vykazoval klesající trend s rostoucím počtem shluků. Relativně vyšší hodnota byla pozorována u 10 shluků, poté prudce klesá. To signalizuje, že rozdělení na 10 klastrů je efektivní z hlediska meziklastrové variability.

Závěrečné rozhodnutí o volbě 10 klastrů je založeno na kombinované analýze všech tří validačních metrik. Tato volba je podporována všemi třemi metrikami, což naznačuje, že rozdělení na 10 klastrů poskytuje dobrý kompromis mezi vnitřní soudržností shluků a jejich separací.



Obrázek 4.19: Výsledky metrik pro hledání optimálního počtu klastrů u Agglomerative algoritmu.

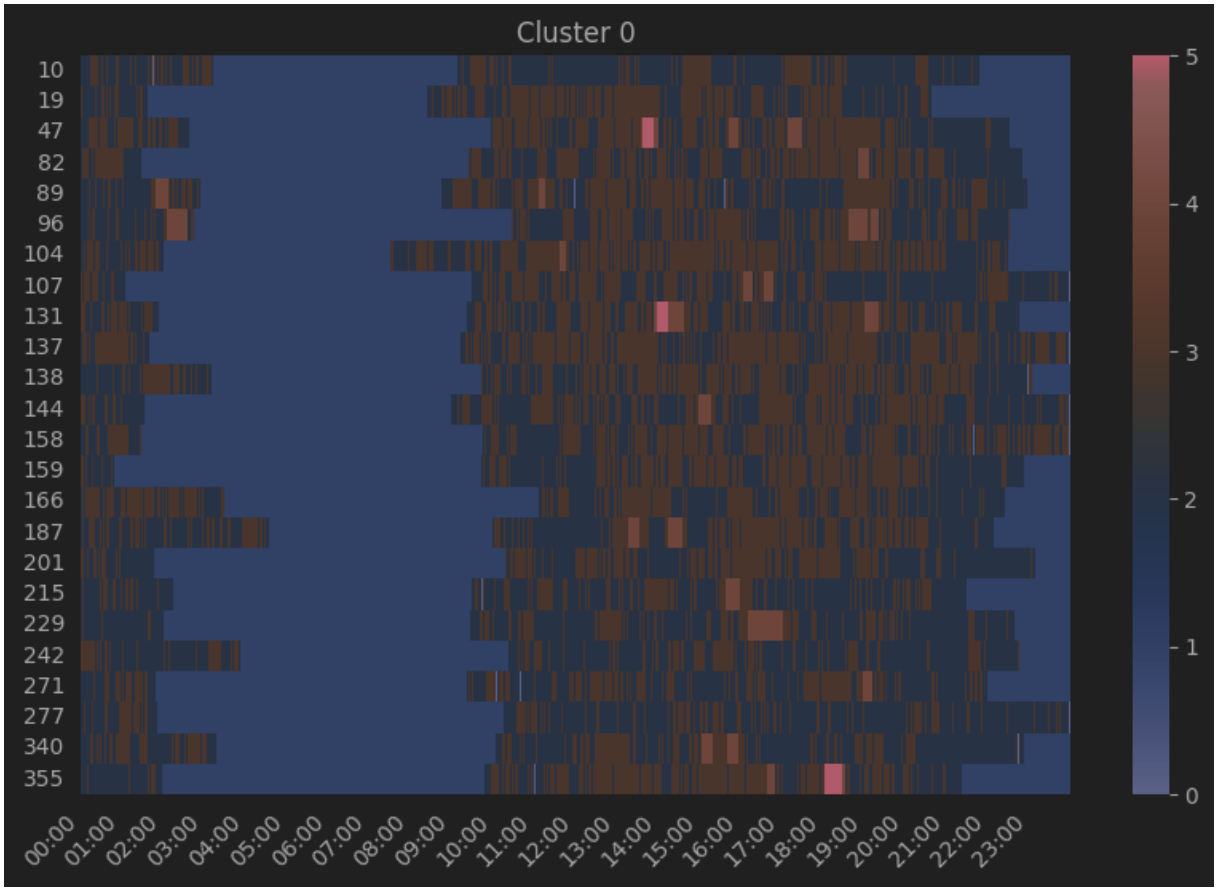
4.1.7 Analýza časového rozložení klastrů Agglomerative algoritmu pomocí heatmapy



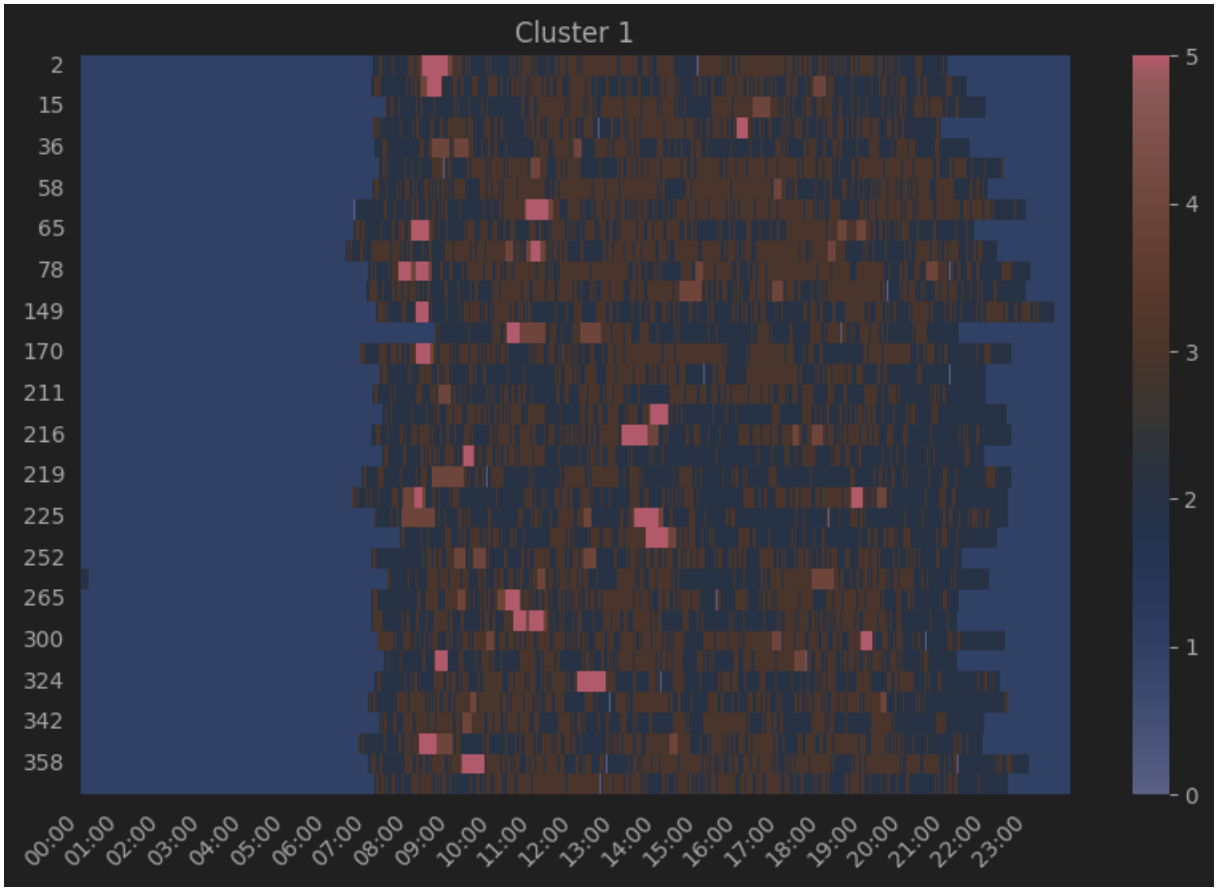
Obrázek 4.20: I u tohoto algoritmu můžeme na levém pruhu vedle dat pozorovat, že klastry se neskládají ze dnů jdoucích bezprostředně po sobě.

4.1.8 Zobrazení jednotlivých klastrů Agglomerative algoritmu

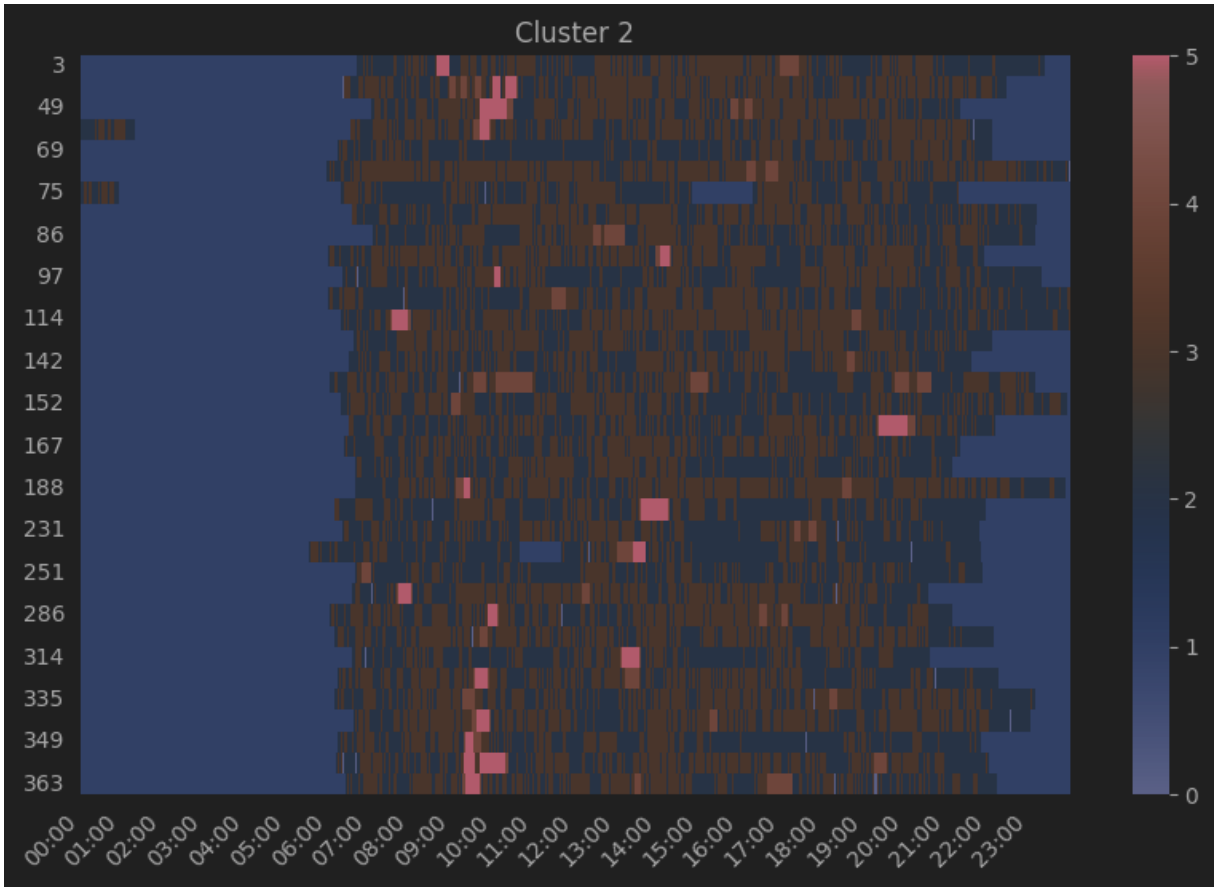
Číslování klastrů není určeno na základě jakýchkoli faktorů, je zcela náhodné, nenaznačuje žádné vlastnosti klastrů ani dat v nich.



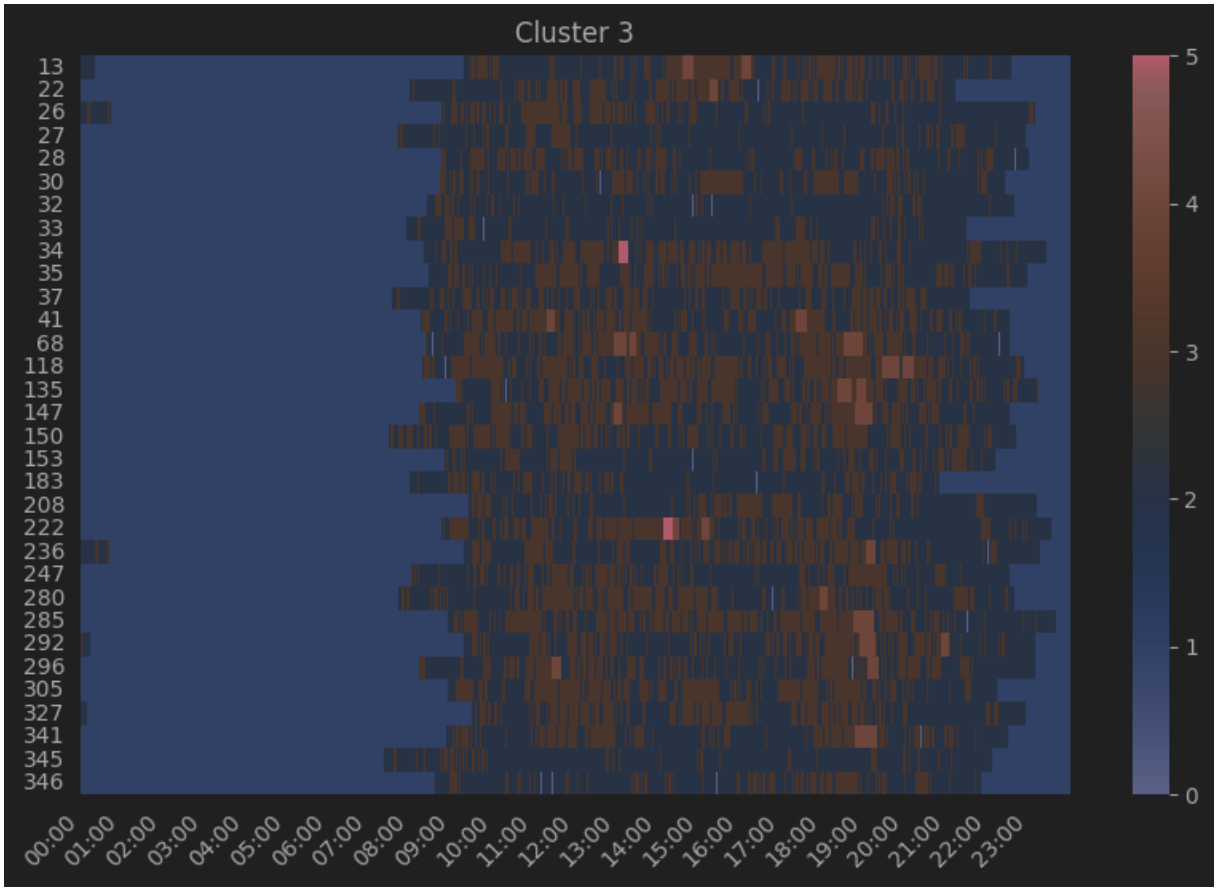
Obrázek 4.21: Klastr 0 zahrnuje dny, kdy byl uživatel aktivní v brzkých ranních hodinách, konkrétně mezi půlnocí a v některých případech až do 4. hodiny ranní. Následně si uživatel šel odpočinout a vstal mezi 10. a 11. hodinou dopoledne. Po zbytek dne, až do 23. hodiny večer, se jeho aktivita pohybovala převážně mezi sedavým chováním a mírnou aktivitou. Tento klastr tak zachycuje dny s noční aktivitou a klidnějším průběhem dne.



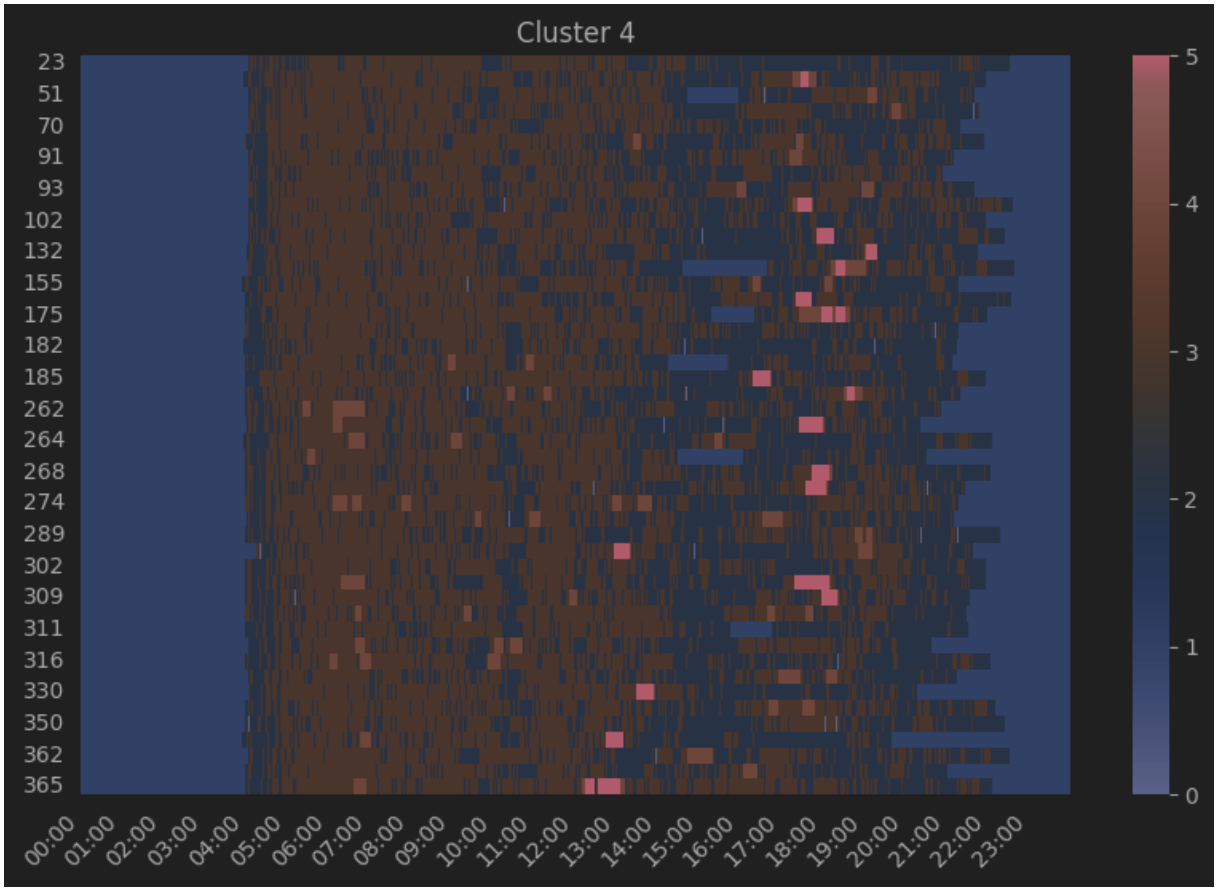
Obrázek 4.22: Shluk 1 zahrnuje dny, kdy uživatel vstával v 8 hodin ráno. Během dne převládala mírná aktivita, která se pravidelně střídala se sedavým chováním. Ojediněle se objevovaly epizody vysoké aktivity, jež však nebyly dominantní. Tento klastr tak zachycuje dny s vyváženým mixem mírné aktivity a odpočinku, doplněné příležitostnými intenzivními aktivitami.



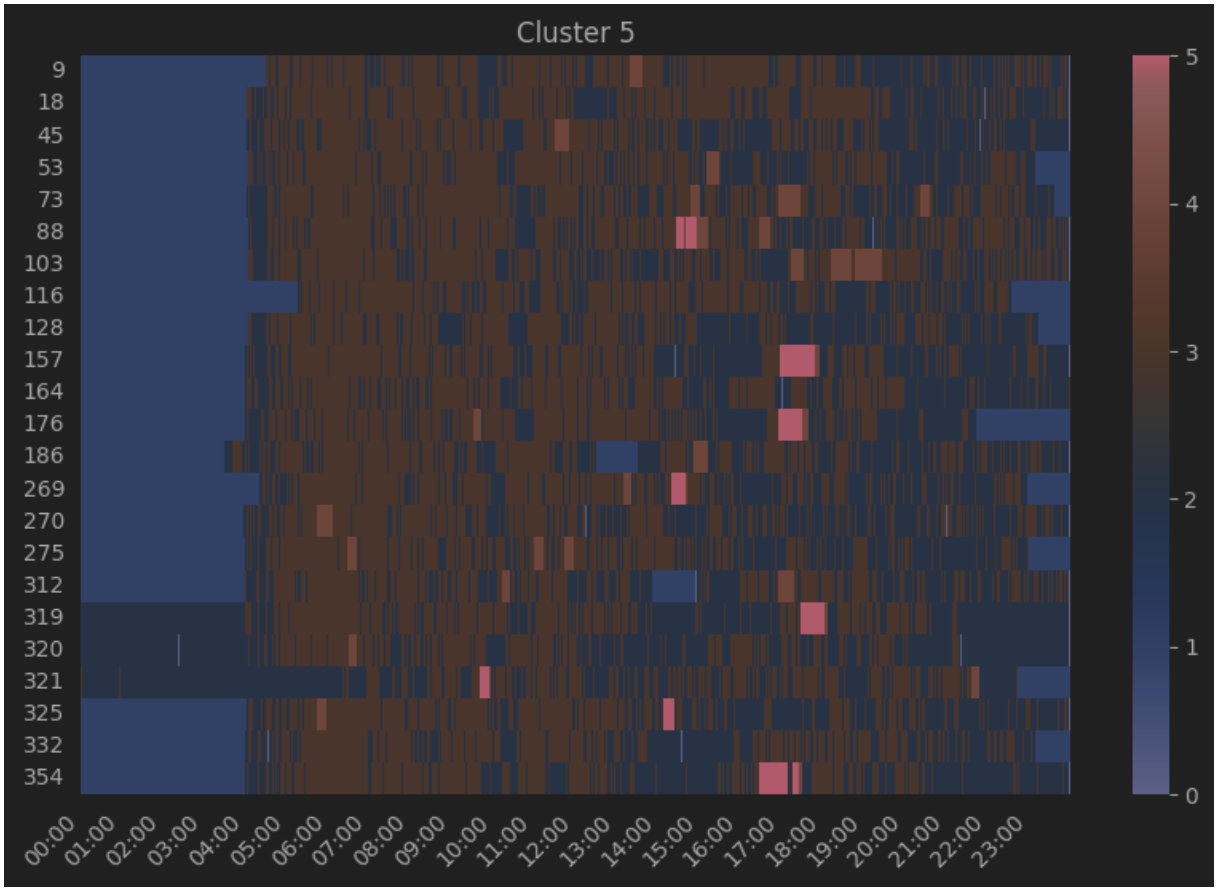
Obrázek 4.23: Klastr 2 zahrnuje dny, kdy uživatel vstával v 7 hodin ráno. Během dne se jeho aktivita pravidelně střídala mezi sedavým chováním a mírnou aktivitou. V ojedinělých případech se objevovala i vysoká aktivita, která však nebyla pravidelná ani dominantní. Tento klastr tak zachycuje dny s vyváženým režimem, kde převažuje klidná až mírná aktivita, doplněná příležitostnými intenzivními momenty.



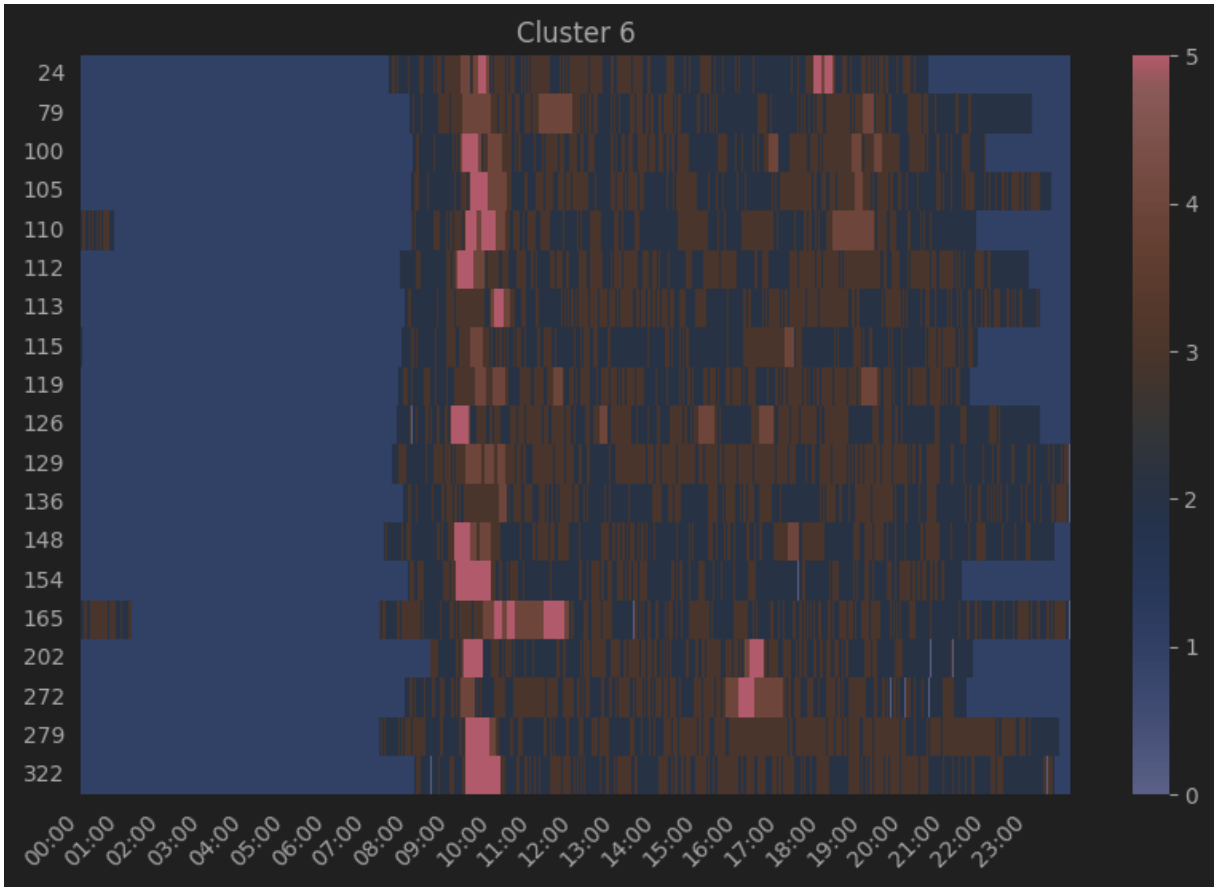
Obrázek 4.24: Klastr 3 zahrnuje dny, kdy uživatel vstával mezi 8. a 10. hodinou ráno. Během dne převažovala sedavost, která se pravidelně střídala s mírnou aktivitou. Hodnoty střední a vysoké aktivity byly zaznamenány pouze velmi zřídka a nebyly dominantní. Tento klastr tak zachycuje dny s převážně klidným režimem, kde mírná aktivita a odpočinek tvoří hlavní charakteristiky dne.



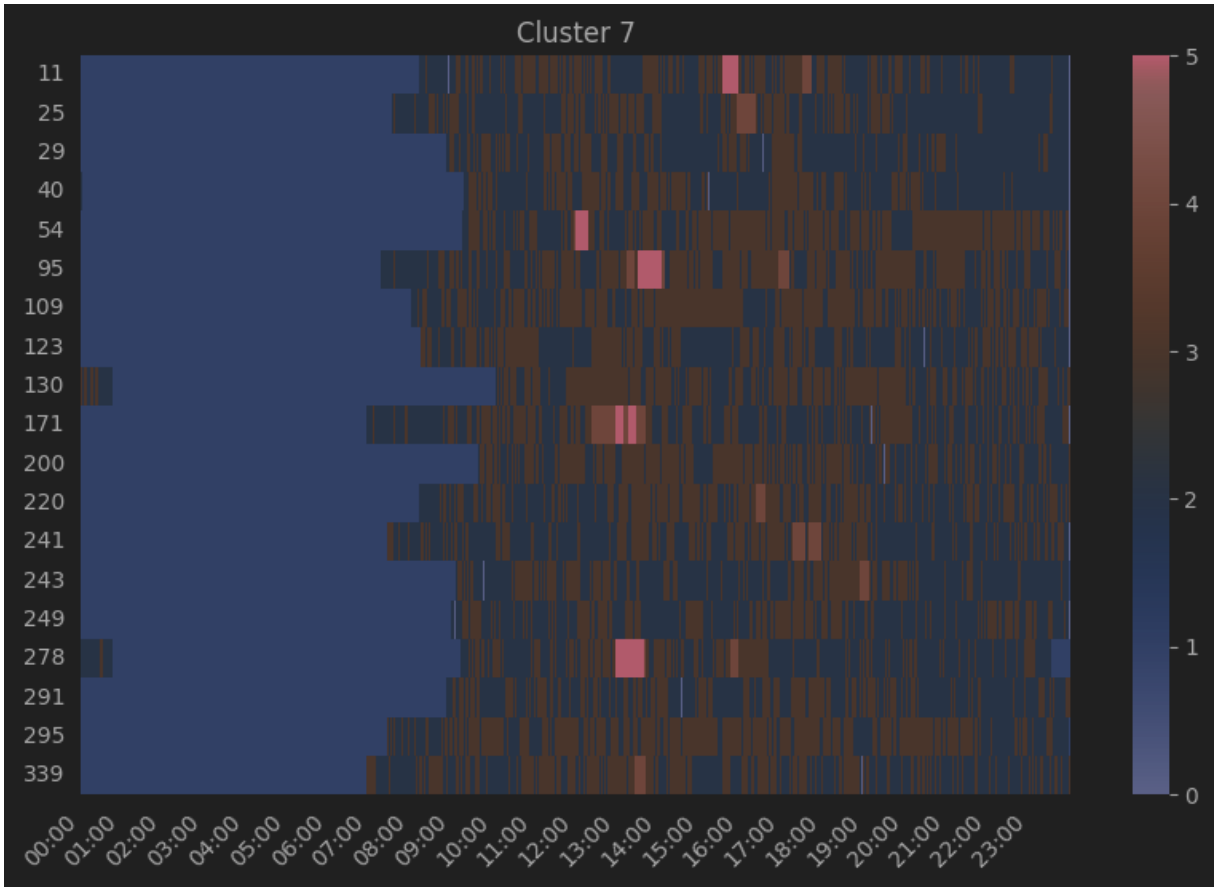
Obrázek 4.25: Shluk 4 zahrnuje dny, kdy uživatel vstával vždy přesně v 4:30 ráno. Následovala souvislá mírná aktivita, jež trvala až do poledních hodin. V odpoledních hodinách převažovalo sedavé chování, avšak v některých dnech se objevily i epizody vysoké aktivity. Tento klastr tak zachycuje dny s výrazným ranním režimem a převážně klidným odpolednem, doplněným příležitostnými intenzivními aktivitami.



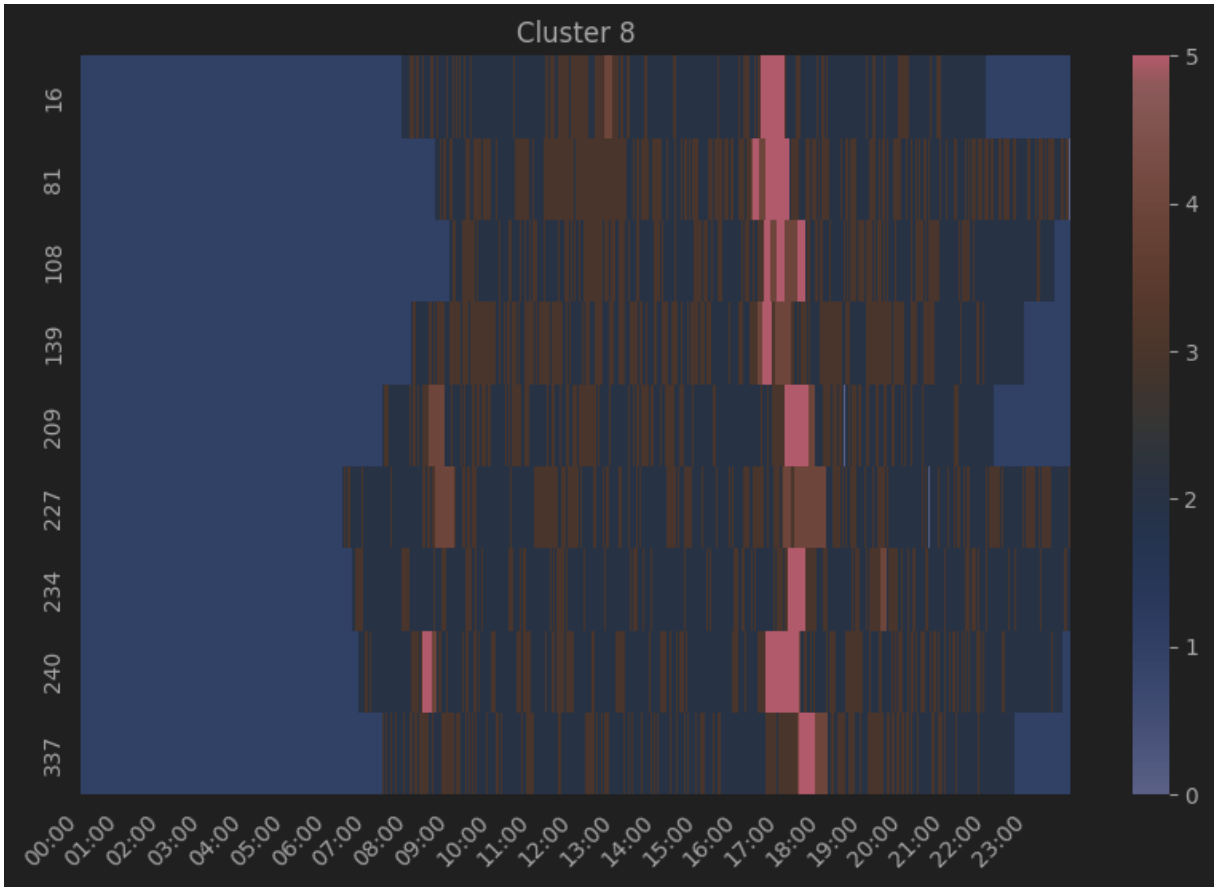
Obrázek 4.26: Shluk 5 zahrnuje dny s podobným režimem jako klastr 4, avšak s jedním klíčovým rozdílem v aktivitě během odpoledních a večerních hodin. Zatímco v klastru 4 uživatel obvykle chodil spát mezi 21. a 23. hodinou, v klastru 5 zůstával aktivní až do půlnoci. Tento rozdíl vyzdvihuje odlišné večerní chování, které odlišuje tyto dva klastry navzdory podobnosti jejich ranního režimu.



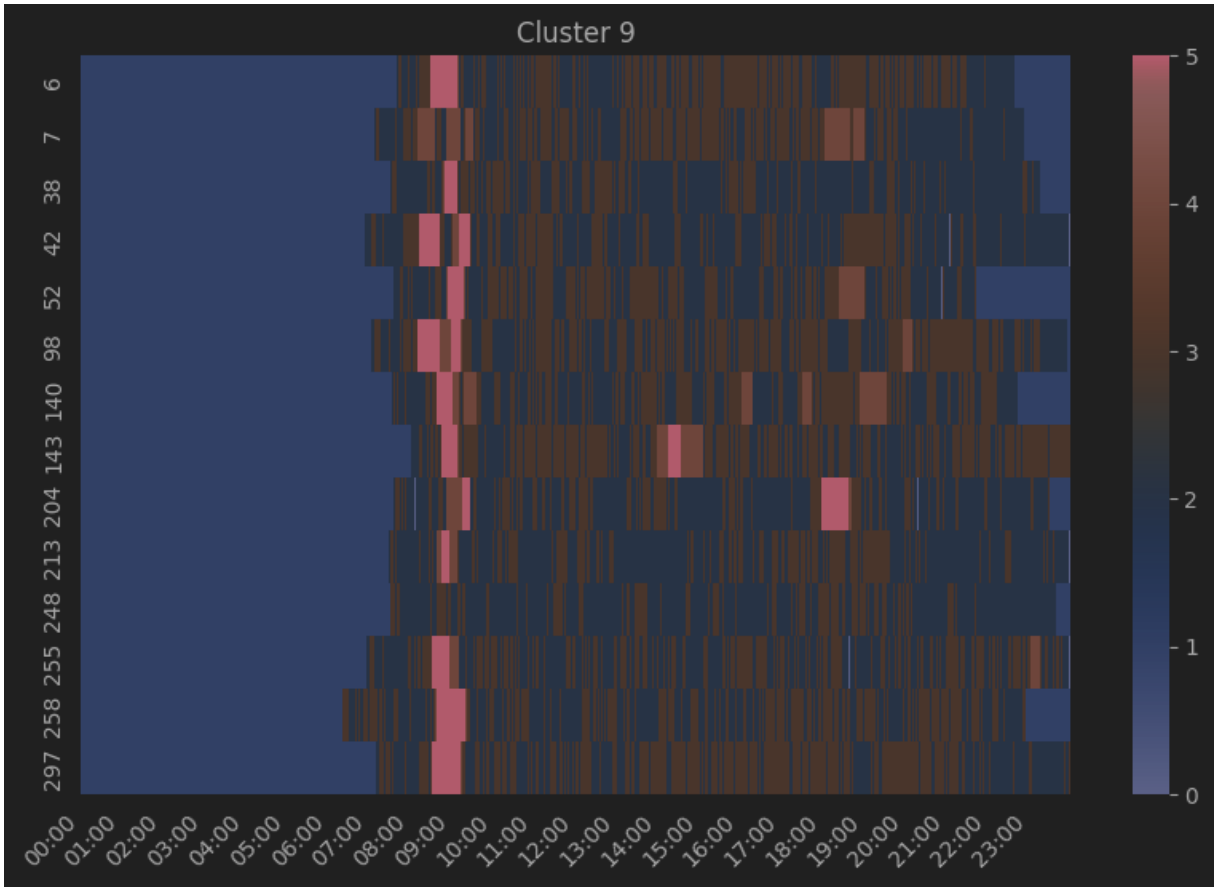
Obrázek 4.27: Klastr 6 zahrnuje dny, kdy uživatel vykazoval vysokou aktivitu krátce po probuzení. Po zbytek dne se jeho aktivita pohybovala převážně mezi sedavým chováním a mírnou aktivitou. Tento klastr tak zachycuje dny s intenzivním začátkem, po němž následoval klidnější průběh dne.



Obrázek 4.28: Shluk 7 zahrnuje dny, kdy uživatel vstával nepravidelně a obvykle nechodil spát před půlnocí. Během dne se jeho aktivita pohybovala převážně mezi mírnou aktivitou a sedavým chováním. Tento shluk tak zachycuje dny s nepravidelným režimem a převážně klidným průběhem dne.



Obrázek 4.29: Klastr 8 zahrnuje dny, kdy uživatel vykazoval pravidelnou vysokou aktivitu kolem 18. hodiny. Tento klastr tak zachycuje dny se specifickým večerním vrcholem aktivity, jenž se opakoval pravidelně.



Obrázek 4.30: Shluk 9 zahrnuje dny, kdy uživatel vykazoval vysokou aktivitu již hodinu po probuzení. Tento klastr tak zachycuje dny s intenzivním začátkem, který následoval krátce po ranním vstávání.

4.2 Základní analýza kategorií

4.2.1 Distribuce hodnot průměrného dne v kategoriích (průměr řádků)

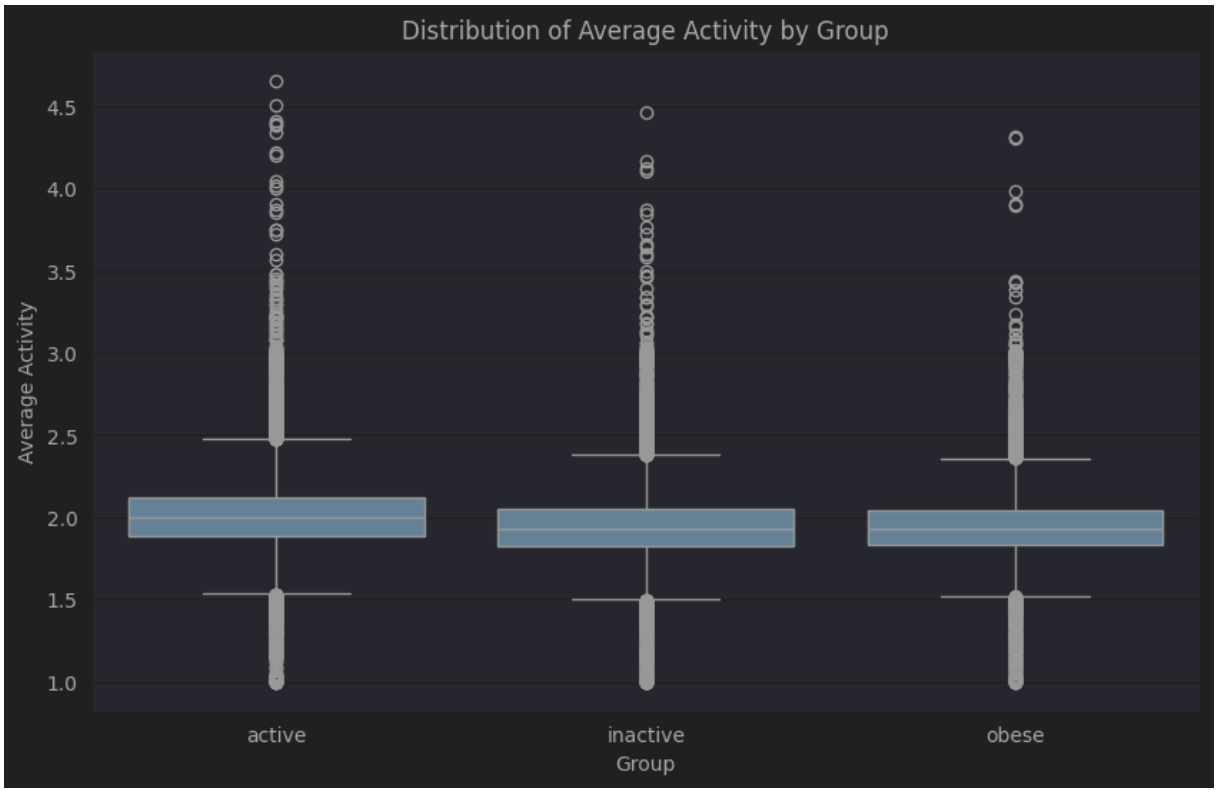
Mediány průměrné aktivity jsou pro všechny skupiny téměř shodné a pohybují se okolo hodnoty 2. Toto zjištění naznačuje, že většina uživatelů ve všech skupinách má podobnou úroveň průměrné aktivity. Rozpětí hodnot

Šířka boxů je ve všech třech skupinách podobná, což ukazuje, že variabilita průměrné aktivity mezi uživateli v jednotlivých skupinách je srovnatelná. To naznačuje, že rozdíly v aktivitě uvnitř každé skupiny nejsou výrazně odlišné. Odlehlé hodnoty

Ve všech skupinách lze pozorovat odlehlé hodnoty, reprezentované kruhy nad a pod boxy. Tyto hodnoty představují uživatele s výjimečně vysokou nebo nízkou úrovní aktivity. Je zajímavé, že odlehlé hodnoty jsou častější u vyšších hodnot průměrné aktivity, což může naznačovat přítomnost vysoce aktivních jedinců. Rozsah dat

Aktivita uživatelů se ve všech třech skupinách pohybuje mezi hodnotami 1 a 4,5. Maximální

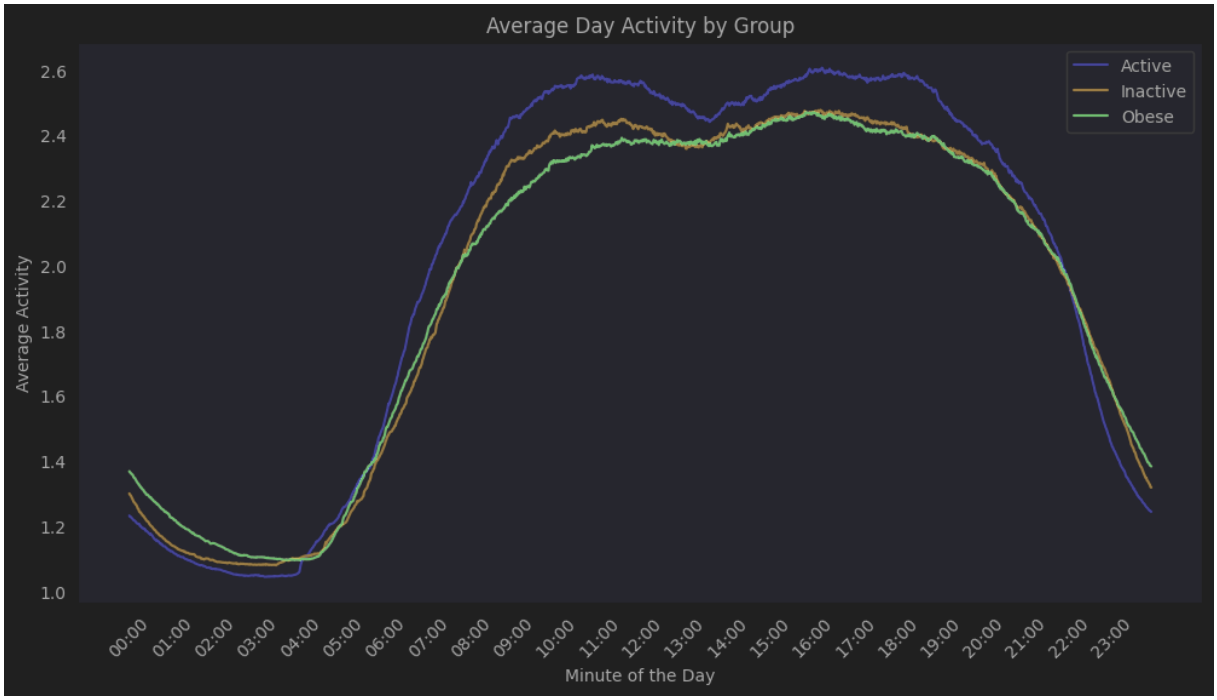
a minimální hodnoty, reprezentované krajními vodorovnými čárami, jsou podobné ve všech skupinách, což potvrzuje konzistenci rozsahu aktivity napříč skupinami.



Obrázek 4.31: Distribuce hodnot průměrného dne v kategoriích (průměr řádků)

4.2.2 Graf průměrného dne kategorie (průměr sloupců)

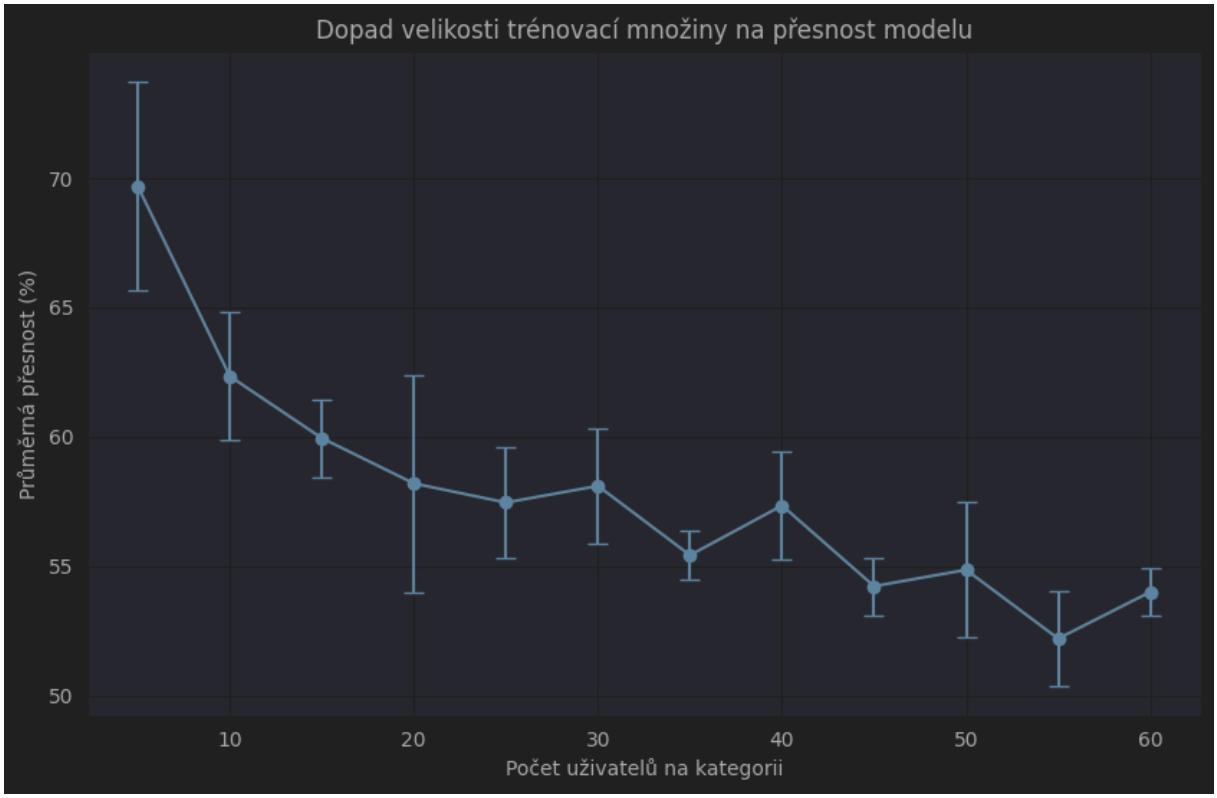
Průměrná aktivita dosahuje svého vrcholu kolem 10:00 hodin. Následuje propad aktivity kolem 13:00 hodin, po němž se aktivita opět zvyšuje na vyšší hodnoty kolem 14:00 hodin. Skupina aktivní vykazuje konzistentně vyšší průměrnou aktivitu ve srovnání se skupinami neaktivní a obézní, což zdůrazňuje rozdíly v úrovni aktivity mezi těmito skupinami.



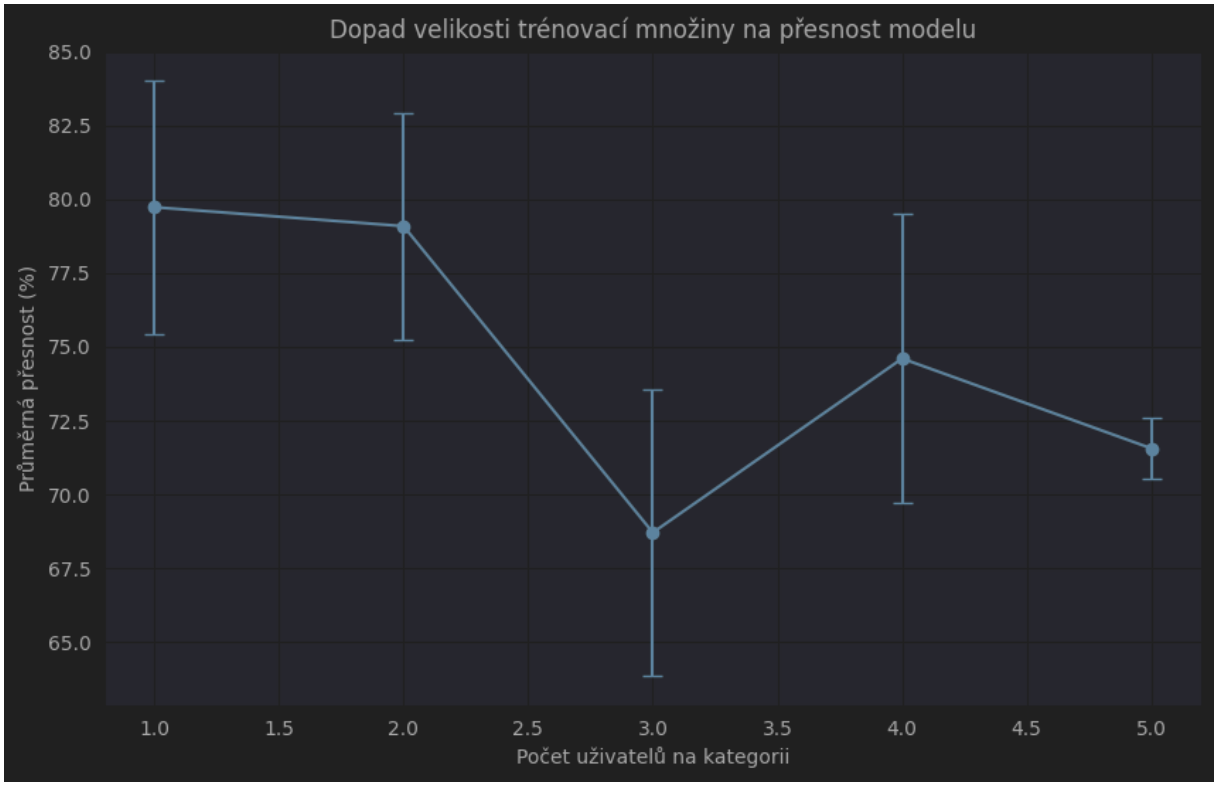
Obrázek 4.32: Distribuce hodnot průměrného dne v kategoriích (průměr řádků)

4.3 Neuronové sítě

4.3.1 Dopad velikosti trénovací množiny na přesnost modelu



Obrázek 4.33: V tomto grafu osa x reprezentuje počet lidí, kteří byli náhodně vybráni z každé kategorie. Osa y představuje průměrnou přesnost. Pro každý počet lidí bylo provedeno 5 pokusů, jejichž výsledky pak byly zprůměrovány. V grafu lze vidět i rozptyl přesností pro danou velikost trénovací sady. Nejlepší výsledky byly naměřeny u nejmenších sad.



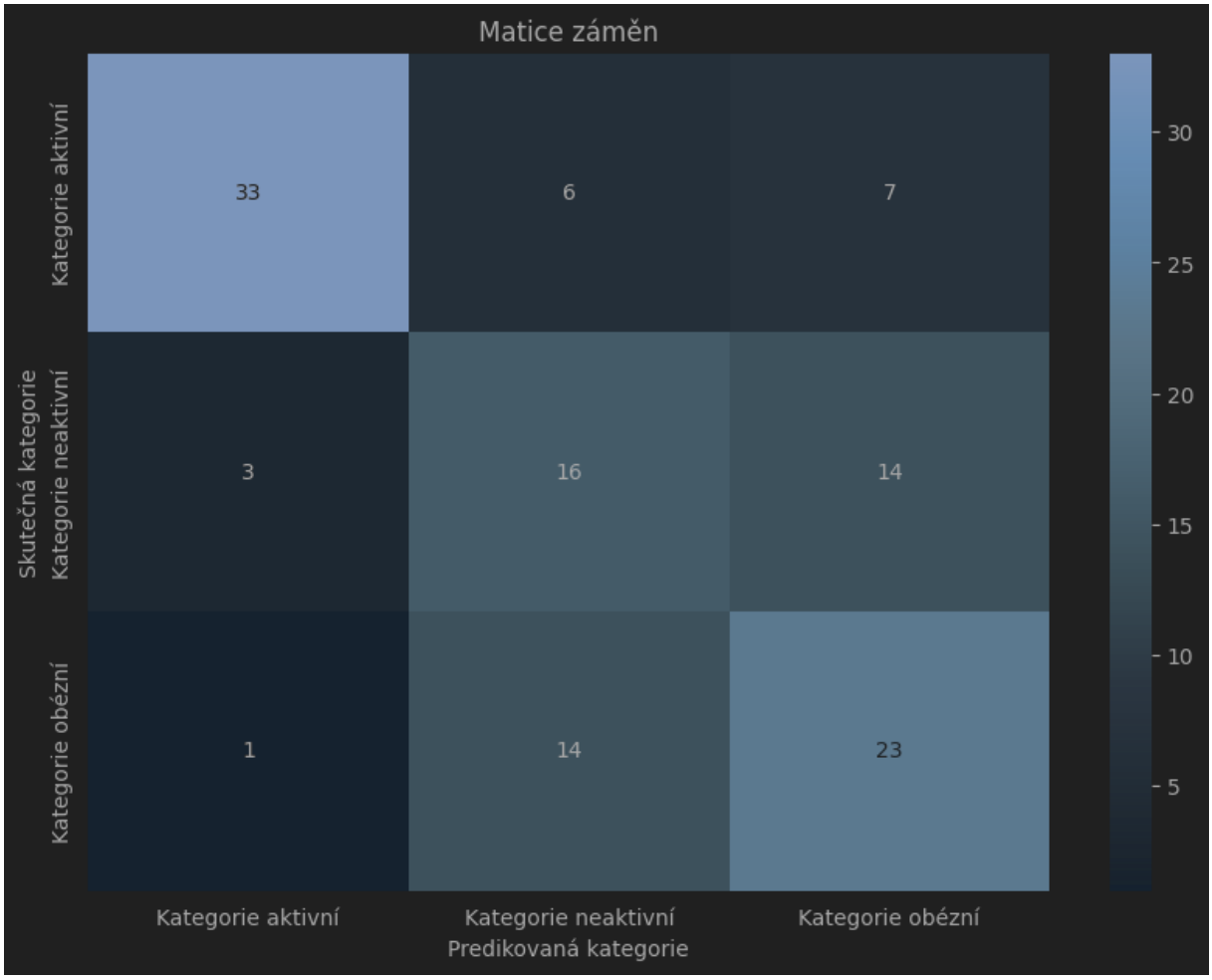
Obrázek 4.34: Na tomto grafu můžeme vidět tento pokus znovu, avšak vybrané velikosti trénovací sady jsou velmi malé. V grafu lze pozorovat nejvyšší přesnost u pouze 1 člověka v kategorii.

4.3.2 Klasifikace uživatelů

V problému klasifikace jednotlivých řádků uživatelů neuronová síť dosáhla 51 %. Když jsem tuto síť použila ke klasifikaci uživatele jako celku, dosáhla 62 %.

Níže zobrazená matice záměn zobrazuje výsledky klasifikace. Řádky reprezentují skutečné kategorie uživatelů, zatímco sloupce ukazují kategorie, do kterých model uživatele zařadil.

Na hlavní diagonále jsou správné predikce. Model správně zařadil 33 uživatelů do kategorie aktivní, 16 uživatelů do kategorie neaktivní a 23 uživatelů do kategorie obézní. Chybné predikce se nacházejí mimo hlavní diagonálu.



Obrázek 4.35: Matice záměn

4.4 Diskuze nad výsledky

4.4.1 Diskuze nad kvalitou klastrů

Vytvořené klastry ukazují jasně odlišitelné vzorce chování uživatele během dne. Kvalita klastrování se projevuje v konzistentních časových vzorcích aktivit v rámci jednotlivých klastrů, což umožňuje efektivní identifikaci typických denních rutin.

4.4.2 Zhodnocení klasifikačního modelu

Model nebyl zvolen dostatečně složitý pro tato nesourodá data, a tak jsou jeho přesnosti při klasifikacích nízké.

Celkově model vykazuje nejlepší výkon při klasifikaci uživatelů do kategorie aktivní, kde má nejvyšší počet správných predikcí a relativně nízký počet chybných. Největší problémy má model s rozlišením mezi kategoriemi neaktivní a ostatními kategoriemi, což je patrné z vyššího počtu chybných zařazení.

4.4.3 Zhodnocení vlivu velikosti trénovací sady na přesnost neuronové sítě

Kvůli složitosti dat se modelu dařilo lépe při zvolení menšího počtu uživatelů. Model byl schopen se naučit menší množství trendů, avšak při velkém množství trénovacích dat selhal rozeznat odlišnosti kategorií.

4.4.4 Potenciální limity použitých metod

Mezi hlavní limity použitých metod patří závislost na kvalitě vstupních dat, kde chybějící nebo nepřesná měření mohou významně ovlivnit výsledky analýzy. Dalším omezením je potenciální neschopnost modelu zachytit komplexní vzorce chování, které se mohou měnit v čase nebo být ovlivněny externími faktory. Model také může být citlivý na odlehlé jenž a nemusí dostatečně zohledňovat individuální charakteristiky uživatelů, které nejsou zachyceny v dostupných datech. Do budoucna by mohla práce pokračovat zaměřením se na optimalizaci architektury neuronové sítě. Tento krok může vést k lepší schopnosti modelu odhalovat dlouhodobé trendy a vzorce v chování, čímž se zvýší kvalita analýzy a predikcí.

5. Závěr

Hlavním záměrem bylo využít metody strojového učení pro analýzu vitálních dat, konkrétně provést klastrovou analýzu pro rozdělení dat podle trendů uživatelů, klasifikovat uživatele do skupin pomocí neuronových sítí a zhodnotit vliv velikosti vstupního datasetu na přesnost klasifikace. V rámci této maturitní práce se podařilo úspěšně splnit většinu stanovených cílů.

Klastrová analýza byla úspěšně provedena s využitím tří různých algoritmů – K-means, DBSCAN a Agglomerativní klastrování. Každý z těchto přístupů přinesl jedinečný pohled na strukturu dat. K-means algoritmus identifikoval 13 distinktivních vzorců chování uživatelů. DBSCAN algoritmus odhalil dva hlavní klastry a skupinu outlierů, poskytujíc pohled na data z perspektivy hustoty. Agglomerativní klastrování rozdělilo data do 10 skupin, nabízejíc hierarchický pohled na strukturu dat. Tyto metody společně umožnily detailní analýzu denních aktivit uživatelů, odhalujíc různé vzorce spánku, sedavého chování a fyzické aktivity.

Při klasifikaci uživatelů pomocí neuronových sítí bylo dosaženo přesnosti 62 % při kategorizaci uživatelů do tří skupin (aktivní, neaktivní, obézní). Tento výsledek, ačkoli není optimální, poskytuje základ pro další výzkum a zlepšení modelu.

Analýza vlivu velikosti trénovacího datasetu na přesnost klasifikace přinesla překvapivé zjištění – model dosahoval lepších výsledků s menšími trénovacími sadami. Toto zjištění naznačuje, že pro daný problém a architekturu sítě může být menší, ale dobře vybraný dataset je efektivnější, než velké množství potenciálně zašuměných dat.

Je třeba zmínit, že ne všechny aspekty práce byly zcela uspokojivé. Hlavním nedostatkem je relativně nízká přesnost klasifikačního modelu, což naznačuje potřebu dalšího ladění a možná i přehodnocení architektury neuronové sítě. Dalším omezením je potenciální neschopnost modelu zachytit komplexní vzorce chování měnící se v čase.

Do budoucna se nabízí několik směrů pro další výzkum: optimalizace architektury neuronové sítě pro lepší zachycení časových závislostí v datech, experimentování s pokročilejšími metodami předzpracování dat pro redukci šumu a zvýraznění relevantních vzorců nebo integrace dalších zdrojů dat pro komplexnější pohled na životní styl uživatelů.

Celkově lze projekt hodnotit pozitivně. Přestože výsledky nejsou definitivní, představují solidní základ pro další výzkum v této oblasti, která má potenciál významně přispět k personalizované zdravotní péči a zlepšení kvality života.

6. Přílohy

6.1 Literatura

[1] Wikipedia (cs). *Cirkadiánní rytmus*. 25. březen 2025. URL: https://cs.wikipedia.org/wiki/Cirkadi%C3%A1nn%C3%AD_rytmus.

[2] Wikipedia. *Cluster analysis*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Cluster_analysis.

[3] GeeksforGeeks. *Data Preprocessing in Machine Learning*. 25. březen 2025. URL: https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/.

[4] GeeksforGeeks. *Principal Component Analysis (PCA)*. 25. březen 2025. URL: https://www.geeksforgeeks.org/principal-component-analysis-pca/.

[5] Wikipedia (cs). *Učení bez učitele*. 25. březen 2025. URL: https://cs.wikipedia.org/wiki/U%C4%8Den%C3%AD_bez_u%C4%8Ditele.

[6] GeeksforGeeks. *Supervised vs Unsupervised Learning*. 25. březen 2025. URL: https://www.geeksforgeeks.org/supervised-unsupervised-learning/.

[7] Wikipedia (cs). *Učení s učitelem*. 25. březen 2025. URL: https://cs.wikipedia.org/wiki/U%C4%8Den%C3%AD_s_u%C4%8Ditelem.

[8] Wikipedia. *K-means clustering*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/K-means_clustering.

[9] IOPscience. *A review of K-means clustering algorithm*. 25. březen 2025. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1873/1/012074/pdf.

[10] GeeksforGeeks. *K-Means Clustering – Introduction*. 25. březen 2025. URL: https://www.geeksforgeeks.org/k-means-clustering-introduction/.

[11] IBM. *K-Means Clustering*. 25. březen 2025. URL: https://www.ibm.com/topics/k-means-clustering.

[12] GeeksforGeeks. *Hierarchical Clustering*. 25. březen 2025. URL: https://www.geeksforgeeks.org/hierarchical-clustering/.

[13] Wikipedia. *Hierarchical clustering*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Hierarchical_clustering.

[14] scikit-learn. *sklearn.cluster.DBSCAN*. 25. březen 2025. URL: https://scikit-learn.org/dev/modules/generated/sklearn.cluster.DBSCAN.html.

[15] Wikipedia. *DBSCAN*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/DBSCAN.

[16] scikit-learn. *sklearn.cluster.HDBSCAN*. 25. březen 2025. URL: https://scikit-learn.org/dev/modules/generated/sklearn.cluster.HDBSCAN.html.

[17] GeeksforGeeks. *Clustering Metrics*. 25. březen 2025. URL: https://www.geeksforgeeks.org/clustering-metrics/.

[18] Wikipedia. *Silhouette (clustering)*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Silhouette_(clustering).

[19] Wikipedia. *Davies–Bouldin index*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index.

[20] Wikipedia. *Calinski–Harabasz index*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Calinski%E2%80%93Harabasz_index.

[21] GeeksforGeeks. *Rand Index in Machine Learning*. 25. březen 2025. URL: https://www.geeksforgeeks.org/rand-index-in-machine-learning/.

[22] Wikipedia. *Rand index*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Rand_index.

[23] Wikipedia. *F-score*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/F-score.

[24] Wikipedia. *Neural network (machine learning)*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Neural_network_(machine_learning).

[25] Wikipedia (cs). *Umělá neuronová síť*. 25. březen 2025. URL: https://cs.wikipedia.org/wiki/Um%C4%9Bl%C3%A1_neuronov%C3%A1_s%C3%AD%C5%A5.

[26] Wikipedia (cs). *Perceptron*. 25. březen 2025. URL: https://cs.wikipedia.org/wiki/Perceptron.

[27] Wikipedia (cs). *Dopředná neuronová síť*. 25. březen 2025. URL: https://cs.wikipedia.org/wiki/Dop%C5%99edn%C3%A1_neuronov%C3%A1_s%C3%AD%C5%A5.

[28] Wikipedia. *Recurrent neural network*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Recurrent_neural_network.

[29] Towards Data Science. *Parameters and Hyperparameters*. 25. březen 2025. URL: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac.

[30] Wikipedia. *Hyperparameter optimization*. 25. březen 2025. URL: https://en.wikipedia.org/wiki/Hyperparameter_optimization.

[31] GeeksforGeeks. *Hyperparameter Tuning*. 25. břez. 2025. URL: https://www.geeksforgeeks.org/hyperparameter-tuning/.

[32] cs.eitca.org. *What does hyperparameter tuning mean?* 25. břez. 2025. URL: https://cs.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/introduction/what-is-machine-learning/what-does-hyperparameter-tuning-mean/.

6.2 Obrázky

6.3 Ukázky