

AutoGluon – nástroj automatizovaného strojového učení

Radek Sušeň



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Radek Sušeň
Osobní číslo: A21279
Studijní program: B0613A140020 Softwarové inženýrství
Forma studia: Prezenční
Téma práce: AutoGluon – nástroj automatizovaného strojového učení
Téma práce anglicky: AutoGluon – An AutoML Toolbox

Zásady pro vypracování

Vypracujte literární rešerši na téma AutoML (Automated Machine Learning), nástrojů pro AutoML, včetně nástroje AutoGluon.
Vyberte vhodná data pro demonstraci experimentů.
Navrhněte experimenty, na kterých demonstřujete možnosti nástroje.
Proveďte experimenty a výsledky zobrazte a zhodnoťte.
Provedené experimenty okomentujte a zdrojový kód s dokumentací připravte v podobě open source knihovny nebo Jupyter notebooku.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

MASOOD, Adnan, 2021. *Automated Machine Learning:: Hyperparameter Optimization, Neural Architecture Search, and Algorithm Selection with Cloud Platforms*. 1. Packt Publishing. ISBN 9781800565524, 1800565526.
SABHARWAL, Navin a Amit AGRAWAL, 2020. *Up and Running Google AutoML and AI Platform: Building Machine Learning and NLP Models Using AutoML and AI Platform for Production Environment*. 1. BPB Publications. ISBN 9388511921.
LAT, Joshua Arvin, 2022. *Machine Learning Engineering on AWS: Build, scale, and secure machine learning systems and MLOps pipelines in production*. 1. Packt Publishing. ISBN 1803247592.
MCKINNEY, William, 2017. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd ed. O'Reilly Media. ISBN 1491957662.
ERICKSON, Nick et al., 2020. Autogluon-tabular: Robust and accurate automl for structured data: Robust and accurate automl for structured data. *ArXiv preprint arXiv:2003.06505*.

Vedoucí bakalářské práce: **Ing. Adam Ulrich**
Ústav matematiky

Datum zadání bakalářské práce: **25. července 2025**
Termín odevzdání bakalářské práce: **22. srpna 2025**



doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. srpna 2025

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.
- Prohlašuji, že při tvorbě této práce jsem použil nástroj generativního modelu AI Chat GPT; <https://chatgpt.com/> za účelem kvalitnější formulace textu. Po použití tohoto nástroje jsem provedl kontrolu obsahu a přebírám za něj plnou zodpovědnost.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato práce se zabývá problematikou strojového učení, nástrojů pro strojové učení a jeho automatizaci, se zaměřením na python knihovnu Autogluon. Cílem provedených experimentů je prozkoumat schopnosti nástroje při řešení problémů klasifikace, regrese a rozpoznávání obrazů. Pro experimenty byly použity různé datové sady s různými typy dat. Výsledné modely byly hodnoceny metrikami, které nástroj AutoGluon nabízí.

Klíčová slova: strojové učení, automatizace, AutoGluon, hyperparametry, model

ABSTRACT

This bachelor's thesis describes machine learning, tools used for machine learning and automation of it. Thesis is targeted on python library called AutoGluon. Goal of experiments done in the practical part is to explore machines capabilities to solve problems of classification, regression and image detection. Experiments used variety of data sets. Final models were evaluated by metrics of AutoGluon itself.

Keywords: machine learning, automatization, AutoGluon, hyperparameters, model

Chtěl bych tímto způsobem poděkovat všem co mě v mém studiu podporovali. Hlavně svým rodičům a sestře. Velký dík také patří panu Ing. Ulrychovi, který mi dával velmi užitečné připomínky jak k psaní této práce, tak i zbytku studia.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 AUTOMATIZOVANÉ STROJOVÉ UČENÍ (AUTOML)	13
1.1 HISTORIE A VÝVOJ AUTOML	13
1.2 ZÁKLADNÍ PRINCIPY STROJOVÉHO UČENÍ	14
1.3 TYPICKÝ WORKFLOW STROJOVÉHO UČENÍ.....	14
1.4 MOŽNOSTI VYUŽITÍ	15
1.4.1 Finance	15
1.4.2 Zdravotnictví	16
1.4.3 Marketing a obchod.....	16
1.4.4 Průmysl	16
1.5 NÁSTROJE PRO AUTOML	16
1.5.1 AutoGluon.....	16
1.5.2 Google AutoML	17
1.5.3 Microsoft Azure AutoML	17
1.5.4 Amazon SageMaker Autopilot.....	17
1.6 MODELÝ.....	17
1.7 DATOVÉ SADY	18
1.8 TYPY ÚLOH.....	19
2 AUTOGLUON.....	21
2.1 ARCHITEKTURA A PRINCIPY FUNGOVÁNÍ.....	21
2.2 HLAVNÍ MODULY AUTOGLUON.....	22
2.3 WORKFLOW PRÁCE S AUTOGLUON.....	23
2.4 PRESETY PRO AUTOGLUON 1.2.0	23
2.5 HYPERPARAMTERY A JEJICH OPTIMALIZACE	24
2.6 HODNOTÍCÍ METRIKY V AUTOGLUON	24
II PRAKTICKÁ ČÁST.....	26
3 CÍL EXPERIMENTŮ	27
3.1 PŘEHLED DATOVÝCH SAD	27
3.1.1 Tabulková data – Mobile data	27
3.1.2 Časově závislá data – Bitcoin history	28
3.1.3 Obrazová data – Fruit dataset.....	28
3.2 TESTOVACÍ PROSTŘEDÍ.....	29
3.3 EXPERIMENT 1 – PRESETY	30
3.3.1 Metodika	30
3.3.2 Preset medium.....	30
3.3.3 Preset good.....	31

3.3.4	Preset high	32
3.3.5	Preset best.....	33
3.3.6	Preset experimental	34
3.3.7	Zhodnocení výsledků	35
3.4	EXPERIMENT 2 - DÉLKA TRÉNOVÁNÍ	36
3.4.1	Metodika.....	36
3.4.2	Rozptyl jednotlivých časových limitů	36
3.4.3	Zhodnocení výsledků	37
3.4.4	Shrnutí	39
3.5	EXPERIMENT 3 - HYPERPARAMETRY	39
3.5.1	Metodika.....	39
3.5.2	Přehled hyperparametrů	40
3.5.3	Hyperparametry 1	40
3.5.4	Hyperparametry 2.....	41
3.5.5	Hyperparametry 3.....	41
3.5.6	Hyperparametry 4.....	41
3.5.7	Zhodnocení výsledků	42
3.6	EXPERIMENT 4 – VLIV ČASU NA HYPERPARAMETRY	43
3.6.1	Metodika.....	43
3.6.2	Zhodnocení výsledků	43
3.7	EXPERIMENT 5 – PREDIKCE CENY BITCOINU	44
3.7.1	Metodika.....	44
3.7.2	Zhodnocení výsledků	44
3.8	EXPERIMENT 6 – ROZPOZNÁVÁNÍ OBRÁZKŮ	46
3.8.1	Metodika.....	46
3.8.2	Zhodnocení výsledků	48
ZÁVĚR.....		50
SEZNAM POUŽITÉ LITERATURY		51
SEZNAM OBRÁZKŮ		53
SEZNAM TABULEK.....		54
SEZNAM PŘÍLOH.....		56

ÚVOD

Cílem této bakalářské práce je seznámit čtenáře s problematikou strojového učení a jeho automatizace pomocí nástroje AutoGluon. V rozsahu této práce je definováno, co to automatizované strojové učení je, čím se zabývá a jaké výhody má jeho automatizace oproti normálnímu přístupu. Pomocí několika experimentů jsou ukázány schopnosti zkoumaného nástroje.

V teoretické části práce je popsáno, jak strojové učení probíhá a které kroky je možné automatizovat. Dále popisuje, jakými úlohami se tento proces zabývá a jaké data je schopen zpracovat. Jsme také seznámeni s různými metrikami, kterými se hodnotí jednotlivé výstupy strojového učení zvané modely.

V praktické části jsou popsány experimenty, které implementují nástroj AutoGluon při trénování modelů pro různé situace za použití několika vstupních datových sad. Každý experiment se sleduje postup vývoje modelů v závislosti na změně vstupních podmínek a jejich konečné výkonosti. Porovnáváním výsledků se také určuje, zda jsou modely vhodné pro praktické využití.

Toto téma mi bylo navrženo panem Ing. Ulrichem po jedné z našich konzultací, kdy ve mně vyvolal zájem o tuto problematiku. Nástroj AutoGluon byl vybrán, protože je na trhu pouze krátce a jeho možnosti nebyly doposud tak dobře známé jako u ostatních nástrojů.

I TEORETICKÁ ČÁST

1 AUTOMATIZOVANÉ STROJOVÉ UČENÍ (AUTOML)

Strojové učení je jedním z nejrychleji se rozvíjejících oborů v informatice. Má za cíl vytvářet algoritmy nebo jejich kombinace, které by se dokázaly učit z námi předaných dat. Na základě tohoto učení je schopný se rozhodovat či provádět predikce, aniž by bylo za potřebí programovat každý jednotlivý krok. [14][17]

Automatizované strojové učení je proces, využívající různé metody a techniky, jejichž úkolem je automatizace jednotlivých fází strojového učení. Typický postup, který je časově náročný, pro tvoření modelů se skládá z několika kroků. Nejprve musí být vybrána vhodná vstupní data a ty následně předzpracovány tak, aby s nimi mohl algoritmus pracovat. Dalším důležitým krokem je výběr algoritmu. Ten musí být schopen naše data zpracovat a vytvořit modely podle našich představ. Pokud má model správně fungovat, musí být při trénování správně laděny hyperparametry. Před nasazením modelu je potřeba také vyhodnotit jeho výsledky k čemuž se využívá různých metrik. [14][17]

V posledních letech můžeme vidět rostoucí význam AutoML. Kvůli velkému zájmu napříč obory je v praxi požadován rychlejší vývoj a nasazení modelů. Pro komerční účely je důležité zkrácení doby pro uvedení produktu na trh. Pro akademické a výzkumné prostředí naopak umožňují automatizaci experimentování a díky tomu rychlejší objevování nových metod. Z tohoto je patrná snaha velkých firem jako Google, Microsoft a Amazon, nabídnout uživatelům AutoML služby, které jsou jednoduché na použití a dávají nám možnost škálování výpočtů a integraci do již existujících systémů. [14][17]

1.1 Historie a vývoj AutoML

Začátky strojového učení sahají až do poloviny minulého století. Jedním z prvních, kdo otevřel diskusi o inteligentních algoritmech, je Alan Turing. První matematický model neuronu, formalizována teoretickým základem neuronových sítí, který položila už v roce 1943 práce McCullocha a Pittse. Machine learning byl definován později Arthurem Samuelem v roce 1959. Ten vyvinul program pro hru dáma. Vysoké očekávání a technologické omezení způsobilo útlum v 70. a 80. letech. [6]

Počátkem 21. století se, kvůli příliš složité a na odborné znalosti náročné workflow ML, začíná formovat koncept AutoML. Automatické optimalizace hyperparametrů je významným milníkem, především bayesovské optimalizace a následný nástup neural architecture search. V dnešní době se z AutoML stal samostatný obor, který našel podporu

jak v akademickém výzkumu, tak i v komerčních nástrojích. Jedná se jeden z nejrychleji rostoucích oborů umělé inteligence.[18][19]

1.2 Základní principy strojového učení

Strojové učení spadá pod obor umělé inteligence, kde se systémům zvyšuje výkon na základě zkušeností a dat. Pro strojové učení existují tři základní přístupy: učení s učitelem, učení bez učitele a zpětnovazební učení. Pro učení s učitelem je zapotřebí, aby tréninková množina obsahovala jak vstupy, tak i výstupy. Model se tedy naučí predikovat výstupy nových vstupních dat. Typickými úlohami jsou regrese a klasifikace. Učení bez učitele se snaží modely naučit vnitřní strukturu neoznačených dat pomocí shlukování nebo redukce dimenzionality. Zpětnovazební učení je proces, při kterém se modely učí optimální strategii. Zpětná vazba nám dává informace o tom, jestli model uspěl či ne. podle toho je odměněn nebo potrestán.[1]

Meta-learning je jednou z klíčových oblastí pro AutoML. Pro řešení nových problémů se využívá zkušeností z předchozích úloh. Je důležitý pro automatizaci výběru algoritmu a nastavení jeho hyperparametrů. Jejich optimalizace je dalším důležitým principem, na kterém AutoML staví. Jejich správným zvolením můžeme zásadně ovlivnit výkonnost modelu. Rychlost učení, hloubka neuronové sítě a počet stromů jsou některé z parametrů, kterými modely disponují. [2]

Kombinace více algoritmů je metoda, která se nazývá ensemble learning. Díky ní je možné dosáhnout vyšší přesnosti a robustnosti. Dále se pro snížení rizika přeučení a zlepšení dalších vlastností modelů používají metody jako boosting, bagging a stacking. Všechny tyto metody dohromady tvoří základní stavební kameny moderních AutoML nástrojů. [4]

1.3 Typický workflow strojového učení

Vytváření modelu se skládá z několika navazujících fází, jejichž sloučení vzniká tzv. machine learning pipeline. Pro tradiční ML je charakteristicky potřeba vysoká míra odbornosti a zkušeností, aby bylo dosaženo dobrých výsledků. Návrh neuronových sítí a ladění hyperparametrů vyžadují stovky experimentů, které mohou být náročné na výpočetní sílu. Z toho vyplývá také velká časová náročnost. Z těchto důvodů vznikají nástroje pro AutoML, které mají za cíl tyto kroky zjednodušit nebo přímo automatizovat. [8][17]

V první fázi sbíráme data a následně je upravujeme dle potřeby. Tyto data velmi často obsahují chyby jako odlehlé prvky, chybějící hodnoty či duplikáty. Úkolem předzpracování je tedy vyčistit a normalizovat data. [8][17]

Následuje výběr vhodných modelů. Výběr závisí na typu řešené úlohy, zpracovávaných datech, jejich velikosti a cílové proměnné. Pro regresní úlohy se vybírá z lineární regrese a boostingových metod. Klasifikační úlohy mají více možností, mezi které patří například logická regrese, náhodné lesy, rozhodovací stromy a neuronové sítě. Automatizace tohoto kroku probíhá tak, že nástroj natrénuje více modelů a poté vybere ten, který dosáhl nejlepších výsledků. [8][17]

Dalším velmi důležitým krokem je ladění hyperparametrů. Zde může dojít k velkým chybám, a to při nedostatečném učení nebo naopak přeučení. Z tohoto důvodu AutoML téměř celý automatizuje a nechává uživateli možnost pouze zvolit si časový limit. Systém poté sám provádí ladění. [8][17]

Když jsou modely natrénovány je nutné provést kontrolu, která se skládá z validace a testování. Často využívanou metodou je cross-validation. Trénování probíhá na množině dat, která je rozdělena na více podmnožin. Na různých částech těchto dat je poté model opakovaně trénován a testován. Cílem tohoto kroku je zjistit výkon modelu a zda nebyl přeučen. [8][17]

Poslední fází workflow je nasazení modelu do produkce. Je důležité v produkčním prostředí monitorovat model z důvodu poklesu jeho výkonu. Tento pokles je způsoben tzv. data driftem, tedy změnou datové distribuce. Je možné tomuto zabránit jejich pravidelným přetrénováním a kontrolou chování. [8][17]

1.4 Možnosti využití

V praxi se AutoML nástroje rozšiřují díky jejich schopnosti řešit reálné problémy v široké škále oborů. Pomocí automatizace předem zmíněných procesů se AutoML stává přístupným i v oborech, kde se odborníci na strojové učení normálně nevyskytovali. [10][11]

1.4.1 Finance

Strojového učení je ve finančním sektoru využíváno již dlouho. Především tedy v oblastech řízení rizika a detekce podvodů. Uplatnění zde nachází hlavně při zjišťování neobvyklých transakcí, budování systému pro credit score a predikci platebních

schopnostech klientů. Dalším způsobem využití AutoML je predikce vývoje akciových trhů[10][11]

1.4.2 Zdravotnictví

I zde můžeme najít uplatnění pro AutoML a to hlavně při diagnostice onemocnění z pacientových záznamů a lékařských obrazů. Je možné jej používat také pro odhad výsledků léčby pacientů podle doby a závažnosti onemocnění. Je ovšem důležité klást důraz na kontrolu dat, která mohou ovlivnit zdraví člověka. [10][11]

1.4.3 Marketing a obchod

Zde našel AutoML mnoho využití hlavně k predikci chování zákazníků, personalizaci reklam a segmentaci trhu. Obchodníci mohou přizpůsobovat nabídky jednotlivým zákazníkům pomocí a automatizace tvorby doporučovacích systémů. Tím zvyšují jejich spokojenost a loajalitu. Je možné jej také použít v e-commerce pro dynamické stanovování cen optimalizaci skladových zásob a predikci poptávky. [10][11]

1.4.4 Průmysl

V tomto prostředí hraje klíčovou roli snížení nákladů na opravy a prostoje ve výrobě. Prediktivní údržba využívá modelů, které předpovídají pravděpodobnost selhání strojů podle senzorických dat. Dále mohou optimalizovat výrobní procesy pomocí snižování spotřeby energie, zlepšení kvality produktů a zvýšením celkové efektivnosti. Je jej také možné použít pro rychlou reakci na změny v provozu díky IoT technologiím. [10][11]

1.5 Nástroje pro AutoML

S rozvojem AutoML je úzce spojena dostupnost nástrojů. Tyto nástroje se snaží uživatelům, kteří nemají hluboké znalosti datové vědy pracovat s modely a využívat je pro vlastní potřeby. I přes společný cíl se v některých oblastech liší od míry automatizace až po zaměření na typy dat.[15][16]

1.5.1 AutoGluon

Jedná se o open-source knihovnu od AWS. Byla navržena tak aby umožňovala co nejvíce zjednodušit tvorbu modelů. AutoGluon provádí všechny kroky automaticky. Podporuje práci s tabulárními daty, textem, obrázky a multimodálními úlohami. Disponuje

také technologiemi jako stacking, bagging a klasifikací s přirozenou adaptací na datové typy.[15][16]

1.5.2 Google AutoML

Je součástí Google Cloud Platform. Umožňuje vytvářet modely bez nutnosti psát kód. Díky využívání NAS automaticky navrhuje architektury neuronových sítí a ty dále optimalizuje pro dané úlohy. Oblíbenost si našel hlavně v oblastech zpracování přirozeného jazyka a obrazu. Díky integraci s ostatními cloudovými službami a rozsáhlým výpočetním zdrojům dosahuje výborných výsledků.[23]

1.5.3 Microsoft Azure AutoML

Nabízí prostředí, kde můžeme tvořit modely a experimentovat s daty, prostřednictvím vizuálního rozhraní nebo programovatelného API. Podpora MLOps principů automatizuje celý životní cyklus modelů. Také ale nabízí možnost nahlédnout do nastavení algoritmů a tím z části ovlivnit proces ladění.[16]

1.5.4 Amazon SageMaker Autopilot

Zásadní myšlenkou je zjednodušit používání AutoML na co nejméně příkazů. Je součástí ekosystému AWS. Podobně jako ostatní nástroje také automatizuje předzpracování dat a následné trénování modelů. Snaží se kombinovat výhody jednoduchosti a transparentnosti. Je velmi dobře škálovatelný, a proto se hodí i pro velké úlohy náročnější na výkon a rychlost.[25]

1.6 Modely

Jsou jádrem každého procesu strojového učení. Při běžném postupu je výběr správného algoritmu na datovém vědci. Korektní výběr ovlivňuje konečnou kvalitu výsledků. Aby byl tento proces usnadněn AutoML jej automatizuje tím, že testuje více modelů a poté vybírá ten, který dosáhl nejlepších výsledků.[5]

Lineární modely jsou jedny z nejjednodušších a nejdéle používaných algoritmů a patří mezi ně například lineární a logická regrese. Nízká výpočetní náročnost a jednoduchá interpretovatelnost jsou velké výhody. Ty vyplývají z předpokladu použití u úloh s přibližně lineárním vztahem mezi vstupními a výstupními daty. Proto také bývají používány jako baseline při porovnávání složitějších metod.[4]

Rozhodovací stromy rozdělují prostor vstupních proměnných podle hierarchie pravidel. Snadno zachycují nelineární vztahy bez potřeby komplikovaného předzpracování. V praxi se ale spíše používají jejich ensemble, protože jsou náchylné na přeučení. [4][5]

Ensemble metody kombinují více stromů pro vyšší přesnost a stabilitu. Patří mezi ně Random Forest anebo boostingové algoritmy jako XGBoost a LightGBM. Rozdíl mezi nimi je ve způsobu použití stromů. Random Forest trénuje více stromů na náhodných datech. Tento princip se jmenuje bagging. Oproti tomu boosting algoritmy vytváří nové stromy, které využívají chyb předešlých modelů. [4][5]

Neuronové sítě dokážou modelovat složité nelineární vztahy a hledat abstraktní reprezentace dat, díky jejich vrstvené architektuře. Z tohoto důvodu je to nejvýznamnější metoda pro úlohy, které zpracovávají obrazy, text a multimodální data. Pro automatizaci hledání nejlepší topologie sítě AutoML využívá Neural Architecture Search. [4][5]

Metoda k-nejbližších sousedů (kNN) hledá nejbližší vzory mezi daty v tréninkové množině. Jsou to velmi jednoduché algoritmy se snadnou implementací ale pro velká data je nevýhodou výpočetní náročnost. Proto se nepoužívá přímo na produkci ale spíše jako srovnávací metoda. [4][5]

Ensemble learning kombinuje tyto modely. AutoML nástroje jako AutoGluon je schopen automaticky sestavovat složené modely a snížit tím riziko nedostatečného výkonu samostatného algoritmu. Jedná se o klíčovou součást AutoML knihoven. [4][5]

1.7 Datové sady

Jsou jedním ze stavebních kamenů procesu strojového učení. Rozsah a kvalita dat přímo ovlivňuje kvalitu a výkonnost výsledných modelů. U tradičního postupu ML je sběr dat, jejich následné čištění a anotace jednou z nejpracnějších kroků přípravy dat. Tento krok, i přes částečnou automatizaci, ovlivňuje kvalita vstupních dat, které AutoML není schopné ovlivnit, a určují míru dosažitelného učení.

Datové sady se dále rozlišuje podle toho, který typ úlohy reprezentují. Pokud je každému vzorku přiřazena jedna z několika předdefinovaných tříd jedná se o klasifikaci. Když data obsahují spojitě hodnoty a mají sloužit k odhadu kvantitativních veličin, jde z pravidla o regresní datovou sadu. Jestliže máme data, pro které je zásadní závislost na čase jako například vývoj ceny kryptoměny, jedná se o časové řady. Posledním typem jsou multimodální úlohy. Ty pracují s datovou sadou, která kombinuje více typů dat dohromady.

Pokud je vyvíjena AutoML knihovna je zapotřebí ji s jinými a k tomu se používají standardizované benchmarkové sady. Benchmarkování je důležité kvůli objektivnosti. Nástroj AutoGluon byl testován na desítkách tabulárních datasetů z platformy OpenML, které jsou veřejně dostupné. Měří se rychlost trénování, přesnost a robustnost.

Pro větší výpovědní hodnotu je lepší využívat firemní nebo doménová data. Ty nejsou veřejně dostupná. AutoML má zde velkou výhodu pro malé firmy, kde není zapotřebí hlubokých znalostí algoritmů, pro rychlé a jednoduché otestování různých přístupů.

Přes to všechno zůstává zásadním faktorem dohled uživatele, který musí zkontrolovat použité datasety. AutoML nám pomáhá při zpracování dat, ale nemůže zajistit jejich validitu a kvalitu.[13]

1.8 Typy úloh

AutoML systémy jsou navrženy tak, aby byly schopny zpracovávat široké spektrum úloh. V praxi běžně vyskytující se úlohy jsou například klasifikace a regrese, predikce časové řady a zpracování obrazu či multimodálních dat. Aby bylo možné aplikovat AutoML nástroj mezi různé domény, musí mít jednotný workflow.

Jednou ze základních úloh je klasifikace. Vstupním vzorkům jsou přiřazeny předdefinované třídy. Pro tyto účely je využíváno ensemble metod a pomocí automatizace je možné jednoduše zkoušet různé klasifikační přístupy, jako lineární logistiku nebo neuronové sítě. Tato úloha je vhodná například pro detekci spamových e-mailů.

Regresní úlohy mají za úkol predikovat spojité hodnoty. Zásadní schopností modelu je zjišťování mezi více proměnnými. Snaží se také o snížení chyby predikce. Díky automatickému schopnosti AutoML trénovat více modelů současně, je znatelně snížen čas potřebný pro nalezení nejlepšího řešení.

Časově závislá data tvoří náročnou úlohu

Velmi často se AutoML používá pro rozpoznávání obrazu, detekci objektů a segmentaci. Kvůli nutnosti znalosti konvolučních neuronových sítí (CNN), patří k těm složitějším úlohám. AutoML toto zjednodušuje použitím Neural Architecture Search. Ten umožňuje generovat pro daný dataset optimální model.

Kombinací různých typů dat dochází k multimodálnímu učení. V rámci jednoho modelu je možné využít jak obrazová data, tak text a tabulární data. Velkého využití se dočkal ve zdravotnictví, kde analýzou záznamů a klinických údajů pomáhá při diagnóze

pacientů. Další obor, který využívá tohoto přístupu, je e-commerce. Zde kombinuje popisky, produktové obrázky a uživatelská hodnocení. Jedním z průkopníků je AutoGluon, který je schopen integrovat více modalit do jedné predikční pipeline. [9][14]

2 AUTOGLUON

Jedná se open-source knihovnu zaměřenou na automatizované strojové učení. Je vyvíjena společností Amazon Web Servis. Vývojáři se zaměřují na to, aby tento nástroj umožňoval tvorbu modelů s velkou výkonností i pro uživatele, kteří mají malé znalosti v oboru strojového učení. Zároveň ale ponechávají možnosti pro pokročilejší uživatele. AutoGluon byl představen roku 2020 a je stále vyvíjen. Mezi nové rozšíření patří možnost zpracovávat multimodální úlohy a predikce časových řad.

Oproti některým ostatním AutoML nástrojům, které mohou být pouze cloudové, je AutoGluon volně dostupný a lokálně provozovatelný. Uživatel díky tomu získává vyšší míru kontroly nad celým procesem. Kladením důrazu na jednoduchost používání je možné spustit trénink modelu pomocí jen několika řádků. Ale zkušený uživatel má stále možnost ladit hyperparametry, zvolit metriky a další pokročilejší funkce, dle vlastní libosti.

Knihovna podporuje mnoho typů úloh. Od klasických jako klasifikace a regrese až po složitější jako zpracování obrazu a integraci více modalit do jednoho modelu. Ve srovnávacích studiích dosahuje výborných výsledků u většiny používaných datasetů. To potvrzuje její robustnost.

Získává velkou pozornost, a to hlavně kvůli kombinaci jednoduchému používání, škálovatelnosti a skvělé výkonnosti. Je možné ji tedy použít jak pro rychlou prototypizaci nebo přímo nasazení do produkčního prostředí.[18][21]

2.1 Architektura a principy fungování

Jedním z hlavních principů, při navrhování této knihovny, byla automatizace celého tvoření modelu počínaje výběrem, trénováním a optimalizace až po pokročilé techniky jako ensemble learning. Autogluon velmi často vytváří meta-modely, které skládá z několika modelů, pomocí metod stacking a bagging. Výsledný prediktor tím zvyšuje svou stabilitu a přesnost.

Bagging metoda agreguje predikce více modelů, které byly trénovány na jiných náhodně zvolených částech trénovacích dat. Snaží se zabránit možnosti přeučení modelu a jeho rozptyl. Další metoda používaná knihovnou AutoGluon je stacking. Ta spočívá ve využívání meta-learner modelu, který se snaží optimálně zpracovat výstupy základních modelů. AutoGluon aplikuje tuto metodu několikrát a tím vzniká vícevrstvý stacking, kde se každá vrstva učí z výsledků té předchozí.

Na výběr, pro zpracování dat, má knihovna velkou řadu různých modelů. V základu volí mezi těmito: LightGBM, CatBoost, XGBoost, k-nejbližších sousedů, rozhodovací stromy, neuronové sítě anebo naivní Bayesův klasifikátor. Nabízí se také možnost pro uživatele zvolit, které algoritmy se mají použít, ale standardně tento proces zastává knihovna automaticky.

Jednou ze silných stránek je podpora úloh využívající multimodální data. AutoGluon toto realizuje tím, že nejprve zpracuje každou modalitu samostatně a jejich výsledky poté, pomocí jednotné pipeline, sjednocuje do predikce. Toto umožňuje efektivní řešení například pro úlohy, které kombinují obrázek produktu s jeho popisem a cenou.[18][21]

2.2 Hlavní moduly AutoGluon

Aby mohl AutoGluon podporovat více datových modalit a s ní spojené velkou škálu úloh, je koncipován jako modulární knihovna. Pro každý typ dat musí být modely optimalizovány jinak a z tohoto důvodu obsahuje několik přednastavených pipeline. Tím je zajištěna automatizace přípravy dat, výběr a trénování modelů.

Nejpoužívanější modul této knihovny je tabular. Díky mnoha modelům a za pomoci vícevrstvého stackingu je schopna dosahovat na těchto datech vysoké přesnosti. V základním nastavení stačí, aby uživatel zadal vstupní data a zvolil cílovou proměnnou. AutoGluon poté všechny další kroky automatizuje.

TimeSeries modul přidává možnost predikce časových řad pomocí kombinace moderních algoritmů strojového učení a tradičních statistických metod. Časová složka je důležitá pro mnoho finančních ukazatelů jako předpověď energetické spotřeby nebo poptávky zboží. Je jej také možné využít pro složitější finanční modely.

Podporuje také Vision a text moduly. Pro lepší výsledky a rychlejší trénink je využíváno před trénovanými a transfer learning. Vhodný při řešení úloh klasifikace obrázků a analýzy textu.

Velmi významný je multimodal modul, který zpracovává různé typy dat. Kde nestačí pouze jeden typ modulu je AutoGluon schopen spojit více datových modalit do jednoho. Výsledky jednotlivých modalit jsou sloučeny ve výchozí prediktor. Nejčastějším příkladem jsou e-commerce a zdravotnictví, kde se často spojují texty s obrázky a časovými záznamy.[22]

2.3 Workflow práce s AutoGluon

AutoGluon byl navržen tak, aby nebylo zapotřebí příliš uživatelských zásahů. Pracovní postup je proto velmi jednoduchý a zabere jen pár kroků jako je příprava dat, trénink modelů, zhodnocení modelů a následná práce s nimi.

Prvním krokem je načtení a jednoduchá příprava dat. Knihovna je schopna automaticky rozeznat s jakým typem proměnné pracuje a poté provede předzpracování dat. Dokáže doplnit některé chybějící hodnoty a převést v textu zadaná čísla na numerické hodnoty. Pro základní připravení dat je možné použít knihovny Pandas nebo NumPy. Pokud je ovšem proměnná zadaná s jednotkou, je vhodné ji ručně odstranit. Nakonec stačí tato data předat AutoGluonu.

Další fází je trénink modelu. Uživatel zavolá metoda `fit()`. V této metodě je potřeba určit pouze cílová proměnná ale může ji doplnit o `preset` nebo časový limit. O vše ostatní už se opět stará sama knihovna. Výběr modelu jeho ladění a `ensembling`. Protože je možné zahájit trénink za použití minimálního kódu urychluje to práci při tvoření výkonných modelů.

Třetí částí je zhodnocení výsledků, pro které AutoGluon nabízí například metodu `leaderboard()`. Podle zvolené metriky nám seřadí a zobrazí všechny natrénované modely. To uživateli umožňuje porovnat výkony modelů mezi sebou a lépe pochopit, jak bylo dosaženo výsledku.

Posledním krokem je použití modelu. Aby model vrátil výsledky pro nová data je volána metoda `predict()`. Knihovna také umožňuje ukládání a načítání již natrénovaných modelů pro opětovné provedení experimentů. Pokud jsme s modelem spokojeni můžeme jej exportovat do produkčního prostředí.[21][22]

2.4 Presety pro AutoGluon 1.2.0

Presety jsou předdefinované konfigurace, díky kterým může uživatel jednoduše podle potřeby zvolit nastavení modelu. Liší se podle zaměření, požadované rychlosti a přesnosti. Jako výchozí nastavení je používán preset `medium`. Ten se zaměřuje na co nejrychlejší natrénování na úkor možného výkonu. Je vhodný pro první prototyp modelu. Pro kompromis mezi rychlostí a přesností je nejlepší použít preset `good`. Výsledné modely mají dobré výsledky, ale jejich trénink stále netrvá příliš dlouho. Preset `high` nabízí vysokou přesnost na úkor horší inference než u předchozích presetů.

Best preset nebere ohled na čas tréninku a snaží se maximalizovat přesnost. Vhodný, pokud chceme model nasadit do produkce. Nový preset verze 1.2.0 je preset experimental. Používá pokročilejší metody jako foundation model pro tabulární data a paralelní strategii fitování. Značně zrychluje trénink zvláště za použití vícejádrového CPU. [21]

2.5 Hyperparametry a jejich optimalizace

Velkým přínosem AutoGluon knihovny je schopnost automatické ladění hyperparametrů. Ty určují celkové chování trénovacího procesu a následný výkon modelu. Mohou ovlivňovat, jak rychle se bude neuronová síť učit z chyb, které udělá, nebo kolik iterací má během tréninku proběhnout. Pokud bychom hyperparametry měli ladit ručně bude nám to trvat mnohem déle než pomocí automatizace. AutoGluon také rozlišuje mezi globálními hyperparametry ovlivňující celý ensemble a specifickými hyperparametry jednotlivých modelů. [21]

Jedním z nejdůležitějších globálních hyperparametrů je `time_limit`, který určuje maximální dobu tréninku. Po jeho uplynutí AutoGluon ukončí fitování modelů. Dalším velmi důležitým je `hyperparameters`, což je slovník algoritmů, které mají být použity. Jako další můžeme použít `auto_stack`. Ten nám umožňuje zapnout nebo vypnout automatiku stacking. Pokud bychom chtěli ovlivnit hloubku stackingu nabízí se nám hyperparametr `stack_ensemble_levels`. Podobně jsme schopni ovlivnit také počet foldů pro bagging. [21]

Jednotlivé modely poté mají specifické hyperparametry, kterým upravujeme přímo jeden model. U modelu LightGBM je možné změnit např. `learning_rate` nebo `num_boost_round`. `Learning_rate` ovlivňuje rychlost učení z chyb a `num_boost_round` určuje maximální počet iterací. Model CatBoost má podobné možnosti. Pro model XGBoost můžeme upravit `max_depth`, který ovlivňuje hloubku jednotlivých stromů. U neuronové sítě nám AutoGluon dává možnost měnit počet skrytých vrstev pomocí `num_layers` nebo `hidden_size`, která mění počet neuronů v dané vrstvě. `N_estimators` pak u Random Forest modelu mění počet stromů v rozhodovacím lese. [21]

2.6 Hodnotící metriky v AutoGluon

Pro správné posouzení kvality modelů je důležitá volba hodnotících metrik. AutoGluon vybírá ze široké škály metrik a automaticky je přizpůsobuje pro dané typy úloh. Pro klasifikační úlohy se jako výchozí metrika používá `accuracy`, což je podíl korektně

klasifikovaných vzorků. Pro zohlednění menšinových tříd je možné použít `balanced_accuracy`. [21]

Regresní úlohy využívá jako výchozí metriku RMSE. Ta zvýrazňuje odlehle hodnoty. Kdybychom chtěli minimalizovat průměrnou chybu a nezajímá nás tolik její velikost je vhodnější použít MAE.

Použité symboly:

n – počet vzorků v testované/trénovací množině.

i – index jednotlivého vzorku (od 1 do n).

y_i – skutečná hodnota cílové proměnné pro i -tý vzorek.

$\hat{y}_i$ – predikovaná hodnota modelu pro i -tý vzorek.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Jedná se o metriku, která není citlivá na velké chyby. Jejím cílem je ukázat průměrnou chybu mezi skutečnými hodnotami a těmi vytvořenými modelem, přičemž je každá hodnota brána jako absolutní.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Podobně jako MAE počítá rozdíl mezi hodnotami, ale místo absolutní hodnoty je umocňuje na druhou. Tím jsou zdůrazněny větší chyby, a proto se často používá pro optimalizaci modelu.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Tato metrika je pouze odmocněná MSE a používá se pro jednodušší interpretaci výsledků. Čím nižší RMSE vychází tím přesnější jsou výsledky modelu.

II PRAKTICKÁ ČÁST

3 CÍL EXPERIMENTŮ

Cílem experimentů je zkoumat chování nástroje AutoGluon, při zpracování různých datových sad a typů úloh. Je zkoumáno, jak se mění kvalita predikcí v závislosti na použitých presetech nebo hyperparametrech a jestli má vliv časové omezení na trénink. Dále jsou také zkoumány možnosti pro časovou řadu vývoje ceny Bitcoinu. V závěrečné části se testují schopnosti AutoGluonu v oblasti klasifikace obrazových dat v prostředí bez využití GPU.

3.1 Přehled datových sad

Jednotlivé datové sady, využity v experimentech byly zvoleny v závislosti na tom, co měl daný experiment zkoumat a na co se zaměřoval.

3.1.1 Tabulková data – Mobile data

Datová sada obsahující informace o mobilních zařízeních. Jsou zde vypsány jak značky telefonů, tak jejich jednotlivé parametry jako kapacita baterie, rozlišení fotoaparátu, procesor a další. Obsahuje přibližně 900 záznamů každý o 15 atributech. Níže je krátká ukázka této sady (Tabulka 1). Celá datová sada je dostupná v elektronické podobě v repositáři (viz Příloha A).

Tabulka 1 Ukázka datové sady pro mobilní data

Company Name	Apple	Apple	Apple
Model Name	iPhone 16 128GB	iPhone 16 256GB	iPhone 16 512GB
Mobile Weight	174g	174g	174g
RAM	6GB	6GB	6GB
Front Camera	12MP	12MP	12MP
Back Camera	48MP	48MP	48MP
Processor	A17 Bionic	A17 Bionic	A17 Bionic
Battery Capacity	3 600mAh	3 600mAh	3 600mAh
Screen Size	6,1 inches	6,1 inches	6,1 inches
Launched Price (Pakistan)	PKR 224 999	PKR 234 999	PKR 244 999
Launched Price (India)	INR 79 999	INR 84 999	INR 89 999
Launched Price (China)	CNY 5 799	CNY 6 099	CNY 6 499
Launched Price (USA)	USD 799	USD 849	USD 899
Launched Price (Dubai)	AED 2 799	AED 2 999	AED 3 199
Launched Year	2024	2024	2024

(zdroj: vlastní)

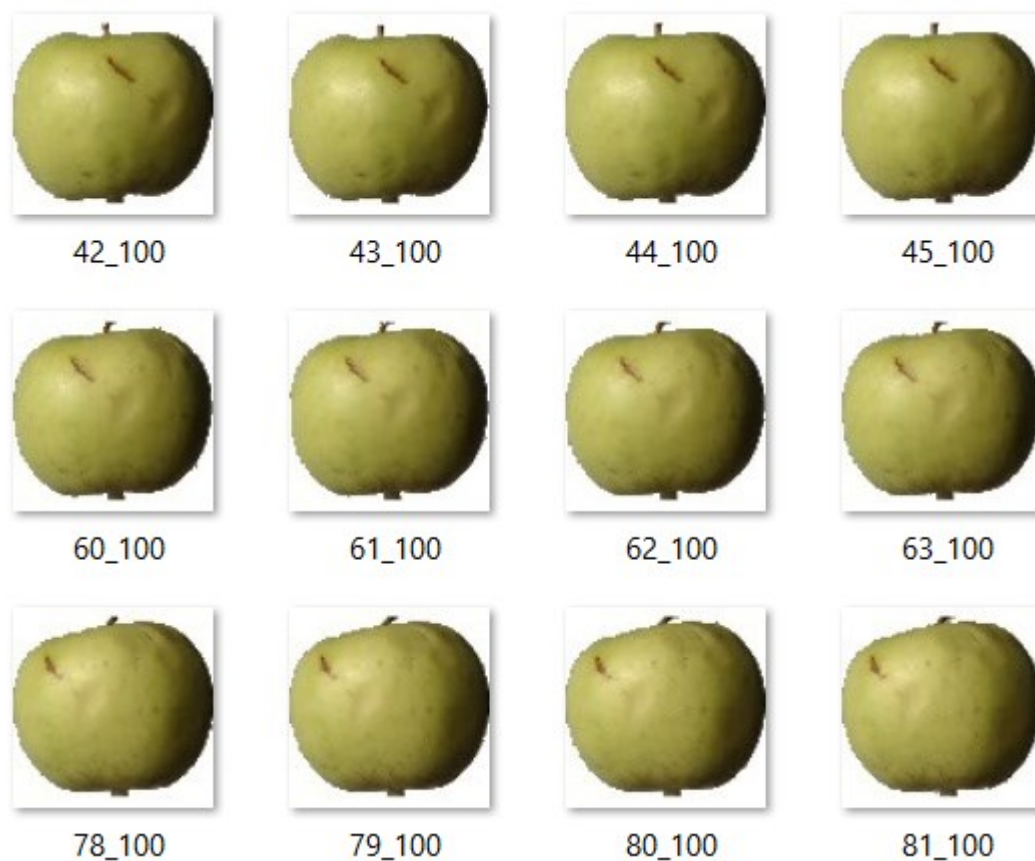
3.1.2 Časově závislá data – Bitcoin history

Pro otestování úlohy typu časová řada byla zvolena datová sada, která zaznamenávala růst a pokles ceny Bitcoinu. Uchovává každodenní průměrnou, maximální, minimální cenu a počet prodaných jednotek od roku 2014 až do roku 2025. Opět je níže krátká ukázka (Tabulka 2) a celá sada je k dispozici v elektronické podobě (viz Příloha B).

Date	Close	High	Low	Open	Volume
17.09.2014	457,33	468,17	452,42	465,86	21056800
18.09.2014	424,44	456,85	413,10	456,85	34483200
19.09.2014	394,79	427,83	384,53	424,10	37919700
20.09.2014	408,90	423,29	389,88	394,67	36863600
21.09.2014	398,82	412,42	393,18	408,08	26580100

3.1.3 Obrazová data – Fruit dataset

Poslední datová sada obsahuje fotografie šesti druhů jablek a je použita pro úlohu klasifikace obrazových dat. Jednotlivé obrázky mají rozměry 100x100 pixelů a zachycují jablka ze všech stran. Obrázky jsou ve formátu .jpg. Celkový počet obrázků této sady je 916 a byla rozdělena na 610 tréninkových obrázků a 306 testovacích. Ukázka datové sady se opět nachází níže Obrázek 1 a celá datová sada je k dispozici v elektronické podobě (viz Příloha C).



Obrázek 1 Ukázka datové sady jablek

(zdroj: vlastní)

3.2 Testovací prostředí

Experimenty probíhaly na stejném zařízení, a to notebooku Acer Aspire 5. Disponuje procesorem Intel Core i3 11. generace a 20 GB RAM paměti a SSD pamětí 238 GB. Pro všechny pokusy bylo použité stejné prostředí.

Tabulka 2 Parametry prostředí

Parametr	Hodnota
Název zařízení	LAPTOP-M4ME25OE
Procesor	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00 GHz (2 jádra)
Paměť RAM	20,0 GB (19,8 GB využitelných)
Úložiště	238 GB SSD NVMe WDC PC SN530
Grafická karta	Intel(R) UHD Graphics (128 MB)

Typ systému	64bitový operační systém, architektura x64
Operační systém	Windows 10
Verze Pythonu	3.10
Verze AutoGluon	1.2.1

(zdroj: vlastní)

3.3 Experiment 1 – presety

První experiment zkoumá možnosti využití AutoGluon knihovny pro trénování predikčních modelů na malé tabulkové sadě. Bylo zjišťováno, jaký vliv na přesnost modelů a dobu trénování, mají jednotlivé přednastavitelné režimy, které nástroj AutoGluon nabízí.

3.3.1 Metodika

- Datová sada: Tabulková sada s parametry mobilních zařízení
- Typ úlohy: Multiklasifikace (103 tříd)
- Cílová proměnná: Launched Price (USA)
- Časové limity: Určeny presety
- Poznámka: Automatická detekce AutoGluonu určila typ úlohy jako multiklasifikační namísto regresečního. To bylo způsobeno tím, že cílový sloupec obsahoval textový řetězec s měnou.

3.3.2 Preset medium

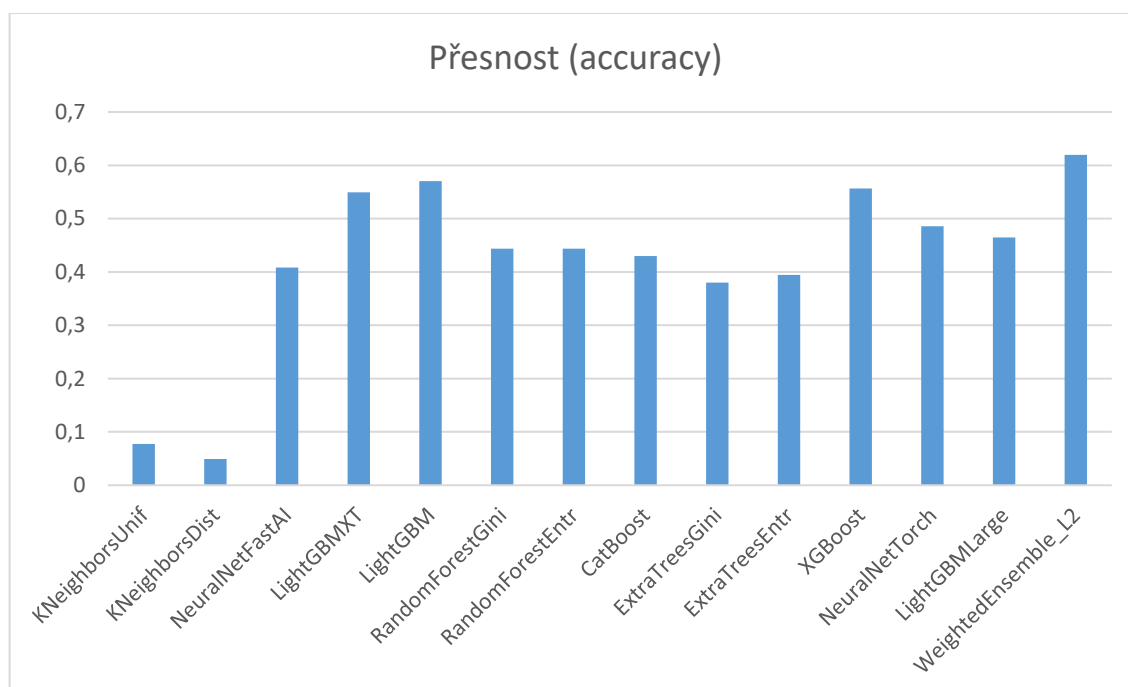
Použitý preset pro trénování nepoužívá stohování ani optimalizaci hyperparametrů. Model s nejlepším výsledkem byl WeightedEnsemble_L2, jehož přesnost byla 0,6197. Je to kombinace výstupů více modelů jako LightGBM, RandomForest a NeuralNetTorch do jednoho. Naopak nízkou přesnost měly modely KNeighbors, které nejsou vhodné pro tento typ úlohy. Model CatBoost měl velmi dlouhou dobu tréninku více než 11 hodin a dosáhl pouze průměrné přesnosti což si můžeme ověřit v tabulce 3.

Tabulka 3 Výsledné hodnoty pro preset medium

Model	Přesnost (accuracy)	Doba trénování (s)	Průchodnost inference (rows/s)
KNeighborsUnif	0,0775	0,01	6201,5
KNeighborsDist	0,0493	0,01	4221,9

NeuralNetFastAI	0,4085	2,89	9096,1
LightGBMXT	0,5493	4,83	1563,2
LightGBM	0,5704	5,07	2173,1
RandomForestGini	0,4437	1,95	3037,4
RandomForestEntr	0,4437	2,2	2990,8
CatBoost	0,4296	41 342,66	9508,2
ExtraTreesGini	0,3803	1,8	3101,5
ExtraTreesEntr	0,3944	1,82	3113,1
XGBoost	0,5563	9,26	3695,1
NeuralNetTorch	0,4859	8,98	11757,8
LightGBMLarge	0,4648	9,5	2228,9
WeightedEnsemble_L2	0,6197	0,11	764,6

(zdroj: vlastní)



Obrázek 2 Graf z tabulky 3

(zdroj: vlastní)

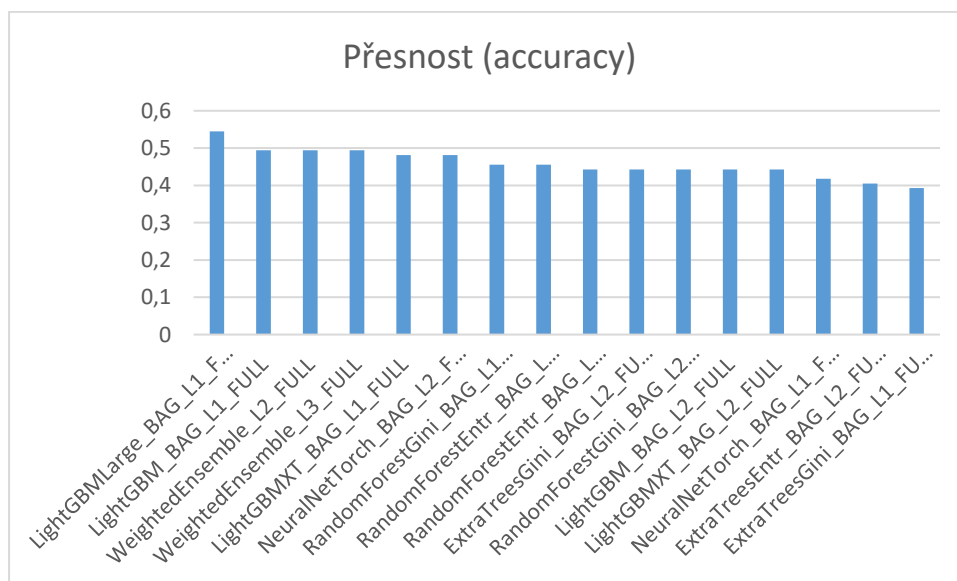
3.3.3 Preset good

Tento preset používá stohování modelů na jedné úrovni. Vytrénoval také více instancí základní modelů. Oproti předchozímu presetu medium, výsledky poukazují na horší přesnost, což je pravděpodobně způsobeno větším zaměřením na generalizaci na úkor výkonu na validačních datech. LightGBMLarge_BAG_L1_FULL dosáhl nejlepších výsledků s přesností 0,5443. Díky stacking a ensemble bylo dosaženo vyšší stability, jak je vidět v tabulce 4. Na obrázku 3 můžeme pozorovat, že rozdíl mezi největší přesností a nejmenší není tak velký jako u medium presetu.

Tabulka 4 Výsledné hodnoty pro preset good

Model	Přesnost (accuracy)	Doba trénování (s)	Průchodnost inference (s)
LightGBMLarge_BAG_L1_FULL	0,5443	6,27	0,470
LightGBM_BAG_L1_FULL	0,4937	2,17	0,177
WeightedEnsemble_L2_FULL	0,4937	2,28	0,188
WeightedEnsemble_L3_FULL	0,4937	2,27	0,189
LightGBMXT_BAG_L1_FULL	0,4810	2,45	0,224
NeuralNetTorch_BAG_L2_FULL	0,4810	15,3	0,822
RandomForestGini_BAG_L1_FULL	0,4557	2,10	0,210
RandomForestEntr_BAG_L2_FULL	0,4557	16,26	0,916
RandomForestEntr_BAG_L1_FULL	0,4430	1,82	0,214
ExtraTreesGini_BAG_L2_FULL	0,4430	14,13	0,802
RandomForestGini_BAG_L2_FULL	0,4430	14,61	0,813
LightGBM_BAG_L2_FULL	0,4430	49,22	0,837
LightGBMXT_BAG_L2_FULL	0,4430	34,86	0,926
NeuralNetTorch_BAG_L1_FULL	0,4177	5,85	0,023
ExtraTreesEntr_BAG_L2_FULL	0,4051	13,96	0,892
ExtraTreesGini_BAG_L1_FULL	0,3924	2,21	0,236

(zdroj: vlastní)



Obrázek 3 Graf z tabulky 4

(zdroj: vlastní)

3.3.4 Preset high

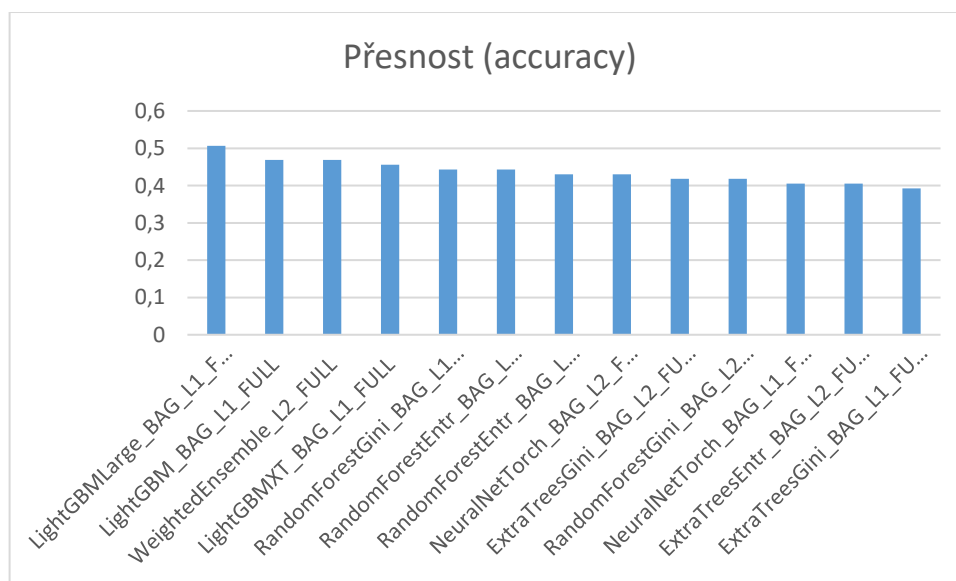
U tohoto presetu byla očekávána vyšší stabilita a přesnost kvůli důkladnějšímu trénování modelů a složitější architekturou ensemble. Některé modely se učily až 50 sekund, a i přes to výsledky poukazují na mírný pokles nebo stagnaci. Některé důvody mohou být

přeučení, nedostatečná kvalita nebo množství tréninkových dat. Nejlepších výsledků opět dosáhl LightGBMLarge_BAG_L1_FULL ale s nižší přesností.

Tabulka 5 Výsledné hodnoty pro preset high

Model	Přesnost (accuracy)	Doba trénování (s)	Průchodnost inference (s)
LightGBMLarge_BAG_L1_FULL	0,5063	6,27	0,421
LightGBM_BAG_L1_FULL	0,4684	2,17	0,203
WeightedEnsemble_L2_FULL	0,4684	2,27	0,214
LightGBMXT_BAG_L1_FULL	0,4557	2,45	0,230
RandomForestGini_BAG_L1_FULL	0,4430	2,10	0,217
RandomForestEntr_BAG_L2_FULL	0,4430	16,26	0,864
RandomForestEntr_BAG_L1_FULL	0,4304	1,82	0,221
NeuralNetTorch_BAG_L2_FULL	0,4304	15,30	0,781
ExtraTreesGini_BAG_L2_FULL	0,4177	14,13	0,840
RandomForestGini_BAG_L2_FULL	0,4177	14,61	0,861
NeuralNetTorch_BAG_L1_FULL	0,4051	5,85	0,025
ExtraTreesEntr_BAG_L2_FULL	0,4051	13,96	0,911
ExtraTreesGini_BAG_L1_FULL	0,3924	2,21	0,243

(zdroj: vlastní)



Obrázek 4 Graf z tabulky 5

(zdroj: vlastní)

3.3.5 Preset best

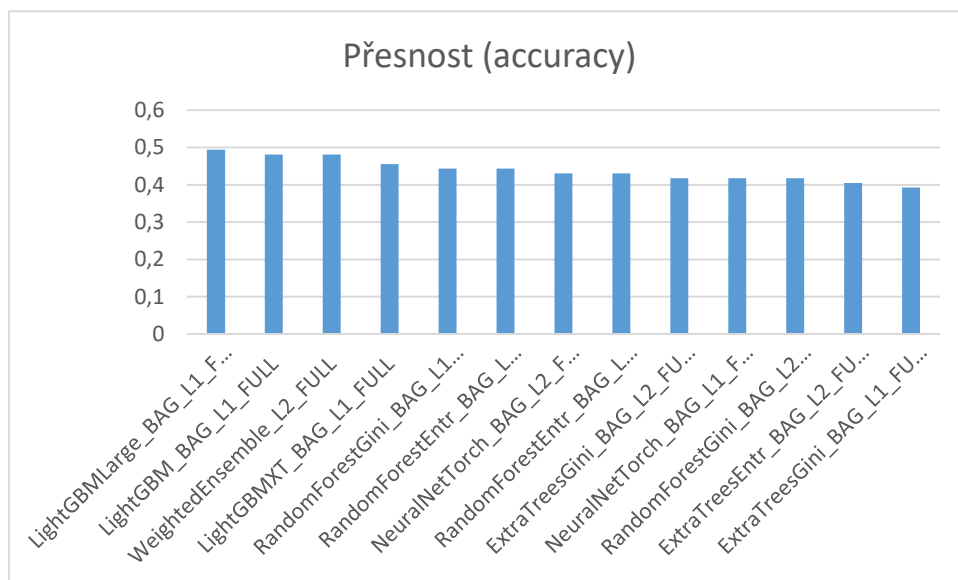
I přes opětovné rozšíření modelového prostoru bylo v pokusu dosaženo nižší nejlepší přesnosti, která dosáhla pouze přesnosti 0,4937 pro model LightGBMLarge_BAG_L1_FULL. Z tabulky je také vychází rozdíl v čase inference pro

modely BAG_L2_FULL a BAG_L1_FULL, kde ani složitější ensemble L2 nepřinesl žádné zlepšení.

Tabulka 6 Výsledné hodnoty pro preset best

Model	Přesnost (accuracy)	Doba trénování (s)	Průchodnost inference (s)
LightGBMLarge_BAG_L1_FULL	0,4937	6,27	0,450
LightGBM_BAG_L1_FULL	0,4810	2,17	0,202
WeightedEnsemble_L2_FULL	0,4810	2,28	0,215
LightGBMXT_BAG_L1_FULL	0,4557	2,45	0,226
RandomForestGini_BAG_L1_FULL	0,4430	2,10	0,214
RandomForestEntr_BAG_L2_FULL	0,4430	16,26	0,884
NeuralNetTorch_BAG_L2_FULL	0,4304	15,30	0,755
RandomForestEntr_BAG_L1_FULL	0,4304	1,82	0,220
ExtraTreesGini_BAG_L2_FULL	0,4177	14,13	0,847
NeuralNetTorch_BAG_L1_FULL	0,4177	5,85	0,024
RandomForestGini_BAG_L2_FULL	0,4177	14,61	0,843
ExtraTreesEntr_BAG_L2_FULL	0,4051	13,96	0,900
ExtraTreesGini_BAG_L1_FULL	0,3924	2,21	0,238

(zdroj: vlastní)



Obrázek 5 Graf z tabulky 6

(zdroj: vlastní)

3.3.6 Preset experimental

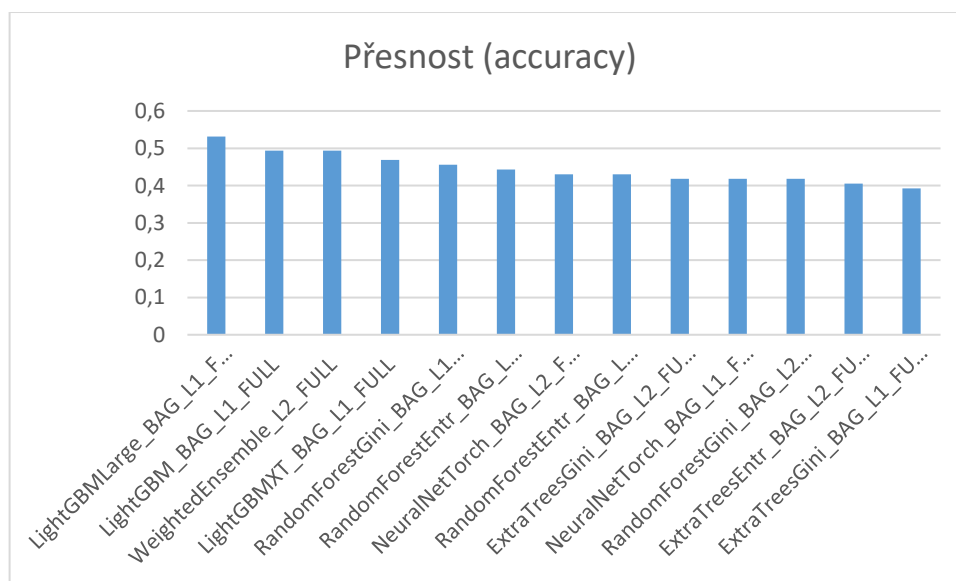
Tento preset cílí na otestování agresivnější konfigurace k čemuž používá rozšířený stackovací režim. Bylo dosaženo druhého nejlepšího celkového výsledku u modelu

LightGBMLarge_BAG_L1_FULL s přesností 0,5316. Ostatní výsledky jsou podobné presetům high a best jak je možné vidět v tabulce.

Tabulka 7 Výsledné hodnoty pro preset experimental

Model	Přesnost (accuracy)	Doba trénování (s)	Průchodnost inference (s)
LightGBMLarge_BAG_L1_FULL	0,5316	6,27	0,415
LightGBM_BAG_L1_FULL	0,4937	2,17	0,201
WeightedEnsemble_L2_FULL	0,4937	2,28	0,212
LightGBMXT_BAG_L1_FULL	0,4684	2,45	0,221
RandomForestGini_BAG_L1_FULL	0,4557	2,10	0,211
RandomForestEntr_BAG_L2_FULL	0,4430	16,26	0,876
NeuralNetTorch_BAG_L2_FULL	0,4304	15,30	0,765
RandomForestEntr_BAG_L1_FULL	0,4304	1,82	0,218
ExtraTreesGini_BAG_L2_FULL	0,4177	14,13	0,830
NeuralNetTorch_BAG_L1_FULL	0,4177	5,85	0,024
RandomForestGini_BAG_L2_FULL	0,4177	14,61	0,835
ExtraTreesEntr_BAG_L2_FULL	0,4051	13,96	0,895
ExtraTreesGini_BAG_L1_FULL	0,3924	2,21	0,235

(zdroj: vlastní)



Obrázek 6 Graf z tabulky 7

(zdroj: vlastní)

3.3.7 Zhodnocení výsledků

Z výsledků experimentu vychází je možné pozorovat, že pro malou datovou sadu s multiklasifikační úlohou je nejvhodnější zvolení nejjednoduššího přístupu použitím presetu

medium. Ukázalo se, že složitější modely nepřinesly očekávané zlepšení a často trpěly na přeučení, kvůli omezenému množství tréninkových dat.

Důležitým poznatkem je také to, že AutoGluon klasifikoval úlohu jako multiklasifikační místo regresní. To bylo způsobeno tím, že cílový sloupec obsahoval data ve formátu textového řetězce kvůli upřesnění měny USD. Důsledkem této chyby je také nižší přesnost, než je u regresní úlohy pro numerické hodnoty očekávané.

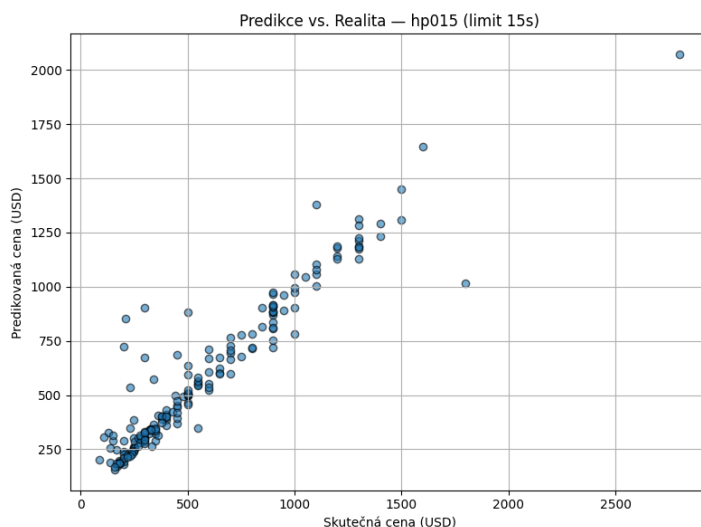
3.4 Experiment 2 - Délka trénování

Experiment měl za cíl porovnat vliv délky trénování na přesnost jednotlivých modelů. Bylo sledováno, zda modely mají dostatek času pro natrénování a jakých výkonů v různých časových limitech dosahují.

3.4.1 Metodika

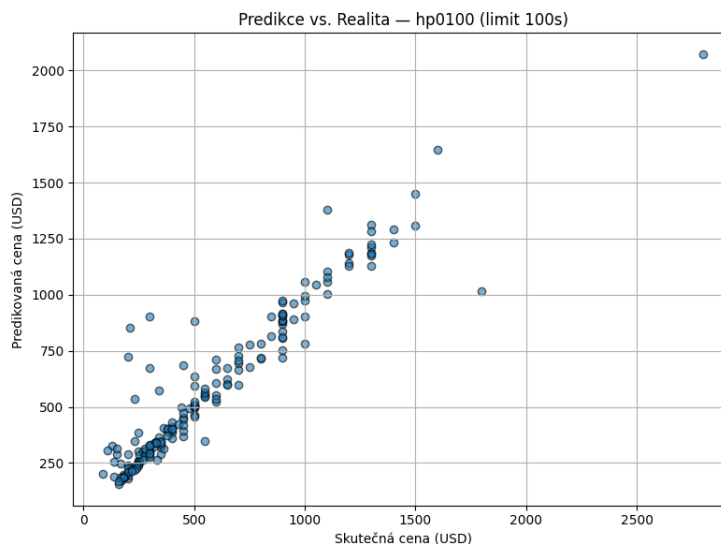
- Datová sada: Tabulková sada s parametry mobilních zařízení
- Typ úlohy: Regrese
- Cílová proměnná: Launched Price (USA)
- Časové limity: 15 s, 100 s, 300 s, 600 s
- Poznámka: Datová sada předzpracována tak, aby cílová proměnná byla numerická hodnota.

3.4.2 Rozptyl jednotlivých časových limitů



Obrázek 7 Rozptyl predikcí pro limit 15 s

(zdroj: vlastní)



Obrázek 8 Rozptyl predikcí pro limit 100 s

(zdroj: vlastní)

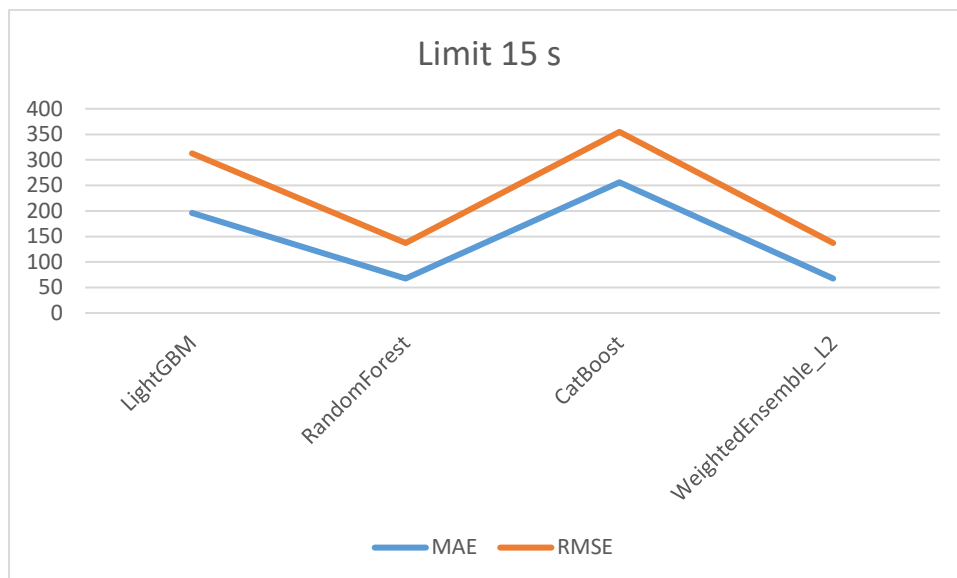
3.4.3 Zhodnocení výsledků

Při nejkratším časovém limitu byl omezen počet natrénovaných modelů, a proto nejsou zahrnuty v tabulce 8, která ukazuje jednotlivé metriky pro použité modely. Z této tabulky také vychází, že WeightedEnsemble_L2 využil pouze RandomForest a dosáhly nejlepších výsledků. Na obrázku 7 vidíme rozptyl predikcí v porovnání se skutečnými hodnotami.

Tabulka 8 Výsledné metriky pro limit 15 s

Model	MAE	MSE	RMSE	R^2	MedianAE	Pearson r
LightGBM	196,35	97 746,65	312,64	0,4030	115,39	0,6657
RandomForest	67,60	18 812,06	137,16	0,8851	25,27	0,9439
CatBoost	256,09	125 930,81	354,87	0,2308	184,70	0,5281
WeightedEnsemble_L2	67,60	18 812,06	137,16	0,8851	25,27	0,9439

(zdroj: vlastní)



Obrázek 9 Graf z tabulky 8 porovnání MAE a RMSE metrik

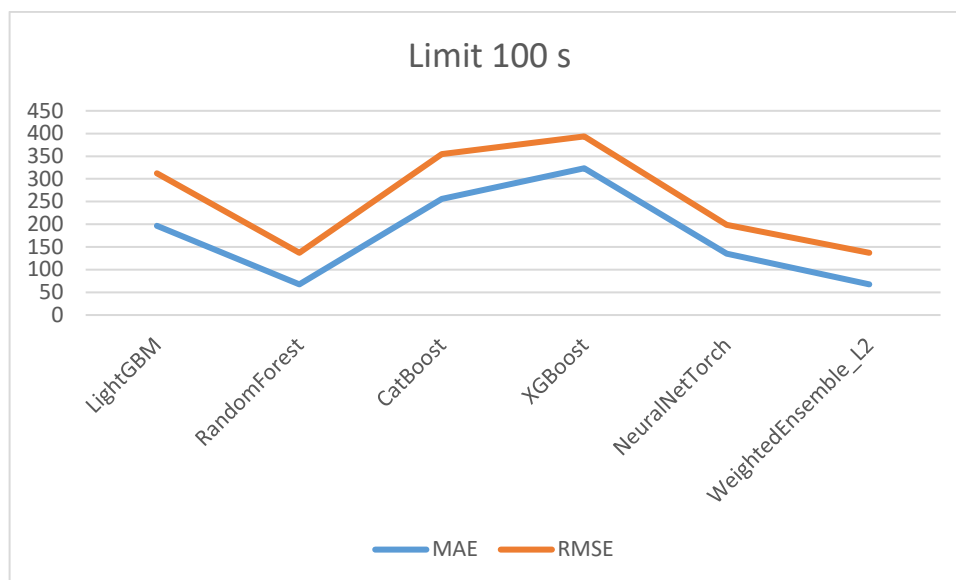
(zdroj: vlastní)

Po prodloužení časového limitu na 100 s už byly natrénovány všechny modely. Nejlepšího výsledku opět dosáhl RandomForest a následně WeightedEnsemble_L2 jak je patrné z grafu na obrázku 10. Protože další prodlužování časového limitu nepřineslo žádné změny oproti tomuto, a proto nejsou zobrazeny v tabulkách ani grafech.

Tabulka 9 Výsledné metriky pro limit 100 s

Model	MAE	MSE	RMSE	R ²	MedianAE	Pearson r
LightGBM	196,35	97 746,65	312,64	0,4030	115,39	0,6657
RandomForest	67,60	18 812,06	137,16	0,8851	25,27	0,9439
CatBoost	256,09	125 930,81	354,87	0,2308	184,70	0,5281
XGBoost	323,42	154 951,79	393,64	0,0536	307,62	0,8443
NeuralNetTorch	135,51	39 389,16	198,47	0,7594	81,93	0,8730
WeightedEnsemble_L2	67,60	18 812,06	137,16	0,8851	25,27	0,9439

(zdroj: vlastní)



Obrázek 10 Graf z tabulky 9 porovnání MAE a RMSE metrik

(zdroj: vlastní)

3.4.4 Shrnutí

Z výsledků je možné usoudit, že i velmi krátký časový limit postačuje k natrénování některých modelů s rozumnou přesností. Po navýšení limitu na 100 s měl AutoGluon čas natrénovat všechny modely ale ani přesto nebyl WeightedEnsemble_L2 schopen překonat výkon RandomForest. Následné navyšování časového limitu již má pouze zanedbatelný až žádný vliv na výkon. Z tohoto se dá usoudit, že datová sada je příliš malá na to aby další trénink přinesl jakékoli benefity.

3.5 Experiment 3 - Hyperparametry

Cílem experimentu je prozkoumat, zda ručně zvolené hyperparametry mohou zlepšit výkon jednotlivých modelů oproti presetu medium. Pro všechny případy byl zvolen časový limit 300 s. Byly zvoleny čtyři různé kombinace hyperparametrů.

3.5.1 Metodika

- Datová sada: Tabulková sada s parametry mobilních zařízení
- Typ úlohy: Regrese
- Cílová proměnná: Launched Price (USA)
- Časové limity: 300 s

- Poznámka: Datová sada předzpracována tak, aby sílová proměnná byl numerická hodnota.

3.5.2 Přehled hyperparametrů

```
"hp1": {
    "GBM": {"learning_rate": 0.03, "num_boost_round": 1000},
    "CAT": {"iterations": 500, "learning_rate": 0.05},
    "XGB": {"max_depth": 8, "learning_rate": 0.05},
    "NN_TORCH": {"num_layers": 2, "hidden_size": 128},
    "RF": {"n_estimators": 300},
},
"hp2": {
    "GBM": {"learning_rate": 0.01, "num_boost_round": 1000},
    "CAT": {"iterations": 400, "learning_rate": 0.05},
    "XGB": {"max_depth": 10, "learning_rate": 0.05},
    "NN_TORCH": {"num_layers": 2, "hidden_size": 128},
    "RF": {"n_estimators": 400},
},
"hp3": {
    "GBM": {"learning_rate": 0.03, "num_boost_round": 800},
    "CAT": {"iterations": 300, "learning_rate": 0.07},
    "XGB": {"max_depth": 8, "learning_rate": 0.03},
    "NN_TORCH": {"num_layers": 3, "hidden_size": 128},
    "RF": {"n_estimators": 300},
},
"hp4": {
    "GBM": {"learning_rate": 0.05, "num_boost_round": 1500},
    "CAT": {"iterations": 700, "learning_rate": 0.05},
    "XGB": {"max_depth": 6, "learning_rate": 0.05},
    "NN_TORCH": {"num_layers": 2, "hidden_size": 128},
    "RF": {"n_estimators": 500},
},
```

(zdroj: vlastní)

3.5.3 Hyperparametry 1

První kombinace zvolena s konzervativními hodnotami pro všechny modely. To je vidět například u NN_TORCH, který má pouze dvě vrstvy s velikostí 128 neuronů. I tak dosáhly tyto hyperparametry solidních výkonů jako RMSE byla 137,16 cpž je jen malé zhoršení oproti presetu medium. MAE stejně jako medián absolutní chyby byly nejnižší ze

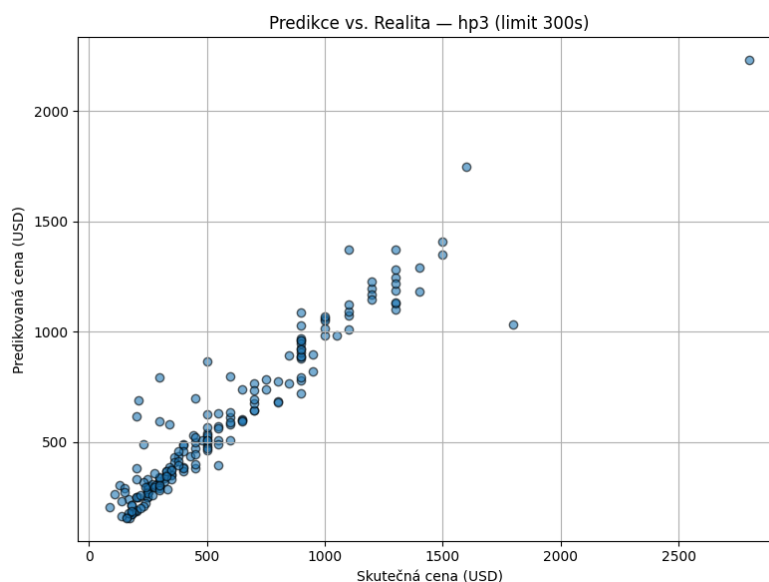
všech kombinací. Ensemble zde preferoval stabilní výstupy RandomForest. Jedná se o velmi vyváženou a spolehlivou volbu pro běžné případy.

3.5.4 Hyperparametry 2

Toto nastavení mělo za cíl otestovat pomalejší učení s větší hloubkou stromových modelů. Pro model GBM byl snížen `learning_rate`, modelu XGB byla zvýšena maximální hloubka stromů a RandomForest bylo dáno více stromů. Podle

3.5.5 Hyperparametry 3

Jednoznačně nejlepších výsledků dosáhla právě třetí kombinace. Při volbě hyperparametrů kladen důraz na rychlejší učení stromových metod a hlubší neuronové sítě. Zvýšením `learning_ratu` bylo zapříčiněno agresivnější učení modelů GBM a CAT. Přidáním vrstvy u NN_TORCH byly zachyceny složitější vzory v datech. RMSE = 125,18 je nejlepší hodnota ze všech kombinací, jak je vidět v tabulce 10. Vidíme také nejnižší MSE, a naopaknejvyšší Pearsonův korelační koeficient 0,9545. Na obrázku 11 proto můžeme vidět, že predikce se drží blízko diagonále.



Obrázek 11 Rozptyl predikcí pro třetí kombinaci

(zdroj: vlastní)

3.5.6 Hyperparametry 4

Poslední kombinace dosáhl celkově nejhorších výsledků. Svým agresivním nastavení měla poskytnout modelům maximální kapacitu. Vysoký počet iterací a počet stromů ani

zvýšený `learning_rate` dokázal právě opačný efekt, než bylo očekáváno. RMSE dosáhl nejvyšší hodnoty 150,73 a naopak Pearsonův koeficient dosáhl nejnižší hodnoty. Jedním z možných důvodů slabých výsledků je pravděpodobně přeučení, a proto tato kombinace pro námi zvolená data není vhodná.

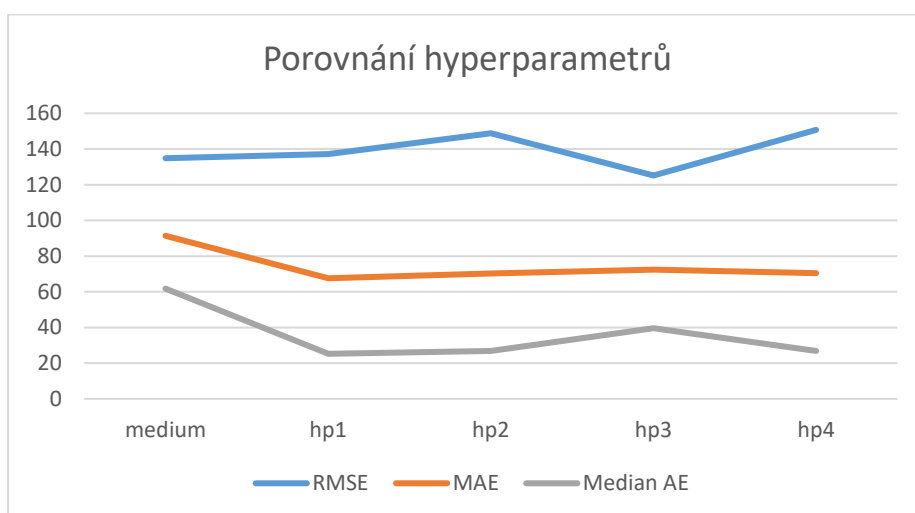
3.5.7 Zhodnocení výsledků

Z výsledků tohoto experimentů je možné vyvodit, že volba hyperparametrů má vliv na kvalitu a výkon modelů. V porovnání s presetem `medium` je vidět, že první kombinace, která využila jednoduché neuronové sítě, dosáhla lepšího MAE. Naopak konfigurace číslo dvě využívající pomalejší, ale hlubšího učení vedla ke zhoršení RMSE. Třetí sada se poté ukázala jako nejúspěšnější, díky mírně agresivnějšímu a zachování hlubší sítě. Poslední varianta, že příliš agresivní učení může vést k přeučení a celkově horším výsledkům. Porovnání těchto kombinací můžeme vidět v grafu na obrázku 12.

Tabulka 10 Porovnání výsledných metrik hyperparametrů

Kombinace	RMSE	MAE	MSE	R ²	Pearson r	Median AE
medium	134,85	91,38	18 185,81	0,8889	0,9533	61,79
hp1	137,16	67,6	18 812,06	0,8851	0,9439	25,27
hp2	148,91	70,22	22 174,70	0,8646	0,9322	26,82
hp3	125,18	72,39	15 670,84	0,9043	0,9545	39,68
hp4	150,73	70,52	22 719,68	0,8612	0,9301	26,94

(zdroj: vlastní)



Obrázek 12 Graf z tabulky 10

(zdroj: vlastní)

3.6 Experiment 4 – vliv času na hyperparametry

Tento experiment má za cíl zkoumat vliv časového limitu na ručně zvolenou kombinaci 1 z předchozího experimentu. Toto nastavení bylo zvoleno, protože dosáhlo dobrých výsledků, které mají možnost se zlepšit.

3.6.1 Metodika

- Datová sada: Tabulková sada s parametry mobilních zařízení
- Typ úlohy: Regrese
- Cílová proměnná: Launched Price (USA)
- Časové limity: 15 s, 30 s, 60 s, 600 s
- Poznámka: Využity hyperparametry v první kombinaci z experimentu 3

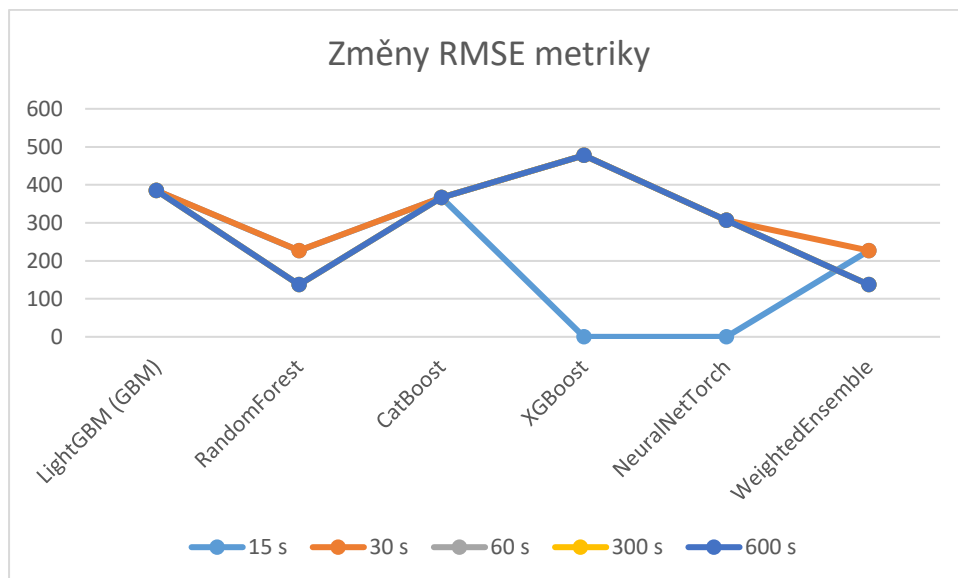
3.6.2 Zhodnocení výsledků

Experiment zkoumal, jaký vliv má časový limit na kvalitu predikcí a výkon výsledného modelu za použití ručně zadaných hyperparametrů. V tabulce 11 vidíme, že při časovém limitu pouze 15 s není možné natrénovat všechny modely. Ale pro RandomForest je to dostatek času, aby dosáhl pěkného výsledku $RMSE = 227,08$, kde po navýšení na 60 s dosáhl ještě nižšího. Ostatní modely však dosahovaly konstantně horších výsledků a ani navýšení na 10 minut nepřineslo žádné zlepšení.

Tabulka 11 Výsledná RMSE hodnota pro experiment 4

Model	15 s	30 s	60 s	300 s	600 s
LightGBM (GBM)	384,69	384,69	384,69	384,69	384,69
RandomForest	227,08	227,08	137,16	137,16	137,16
CatBoost	366,73	366,73	366,73	366,73	366,73
XGBoost	–	477,50	477,50	477,50	477,50
NeuralNetTorch	–	306,56	306,86	306,86	306,86
WeightedEnsemble	227,08	227,08	137,16	137,16	137,16

(zdroj: vlastní)



Obrázek 13 Graf z tabulky 11

(zdroj: vlastní)

3.7 Experiment 5 – predikce ceny Bitcoinu

Experiment zkoumá, jestli je AutoGluon schopen predikovat změny v ceně Bitcoinu. Hlavně tedy prudký nárůst v roce 2024, čímž se zjišťuje, jestli je schopen zachycovat velké změny v trendech na základě historických dat. U tohoto typu úloh je zapotřebí aby měly modely přístup k informacím o předchozích stavech. Proto je potřeba data zpracovat pomocí lag Feature.

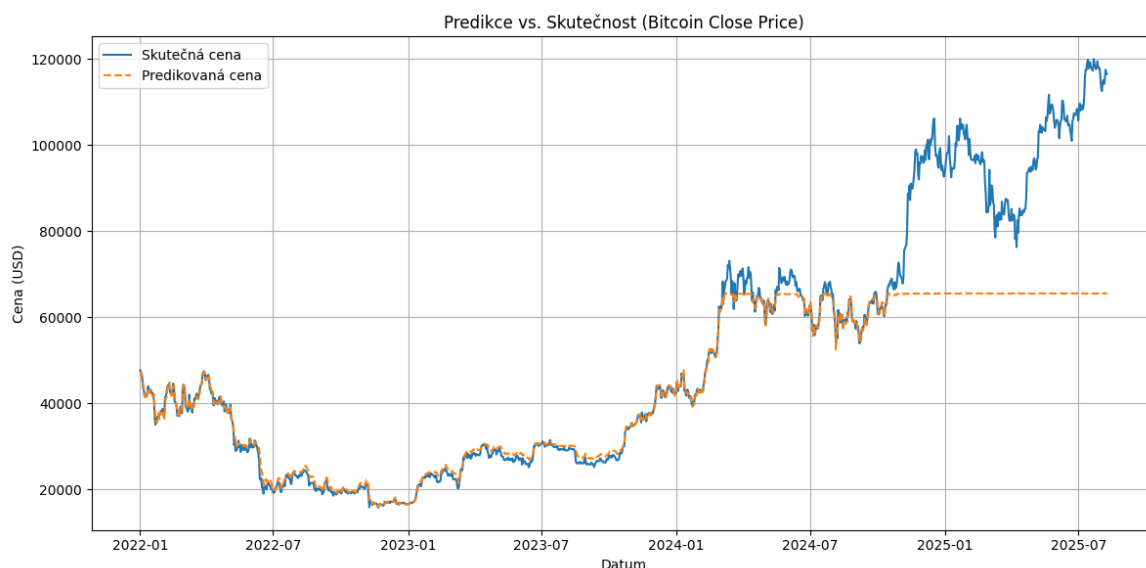
3.7.1 Metodika

- Datová sada: Historie ceny Bitcoinu (viz Příloha B)
- Typ úlohy: Regresní se zaměřením na časovou řadu
- Cílová proměnná: Close (uzavírací cena v daný den)
- Časové limity: 600 s

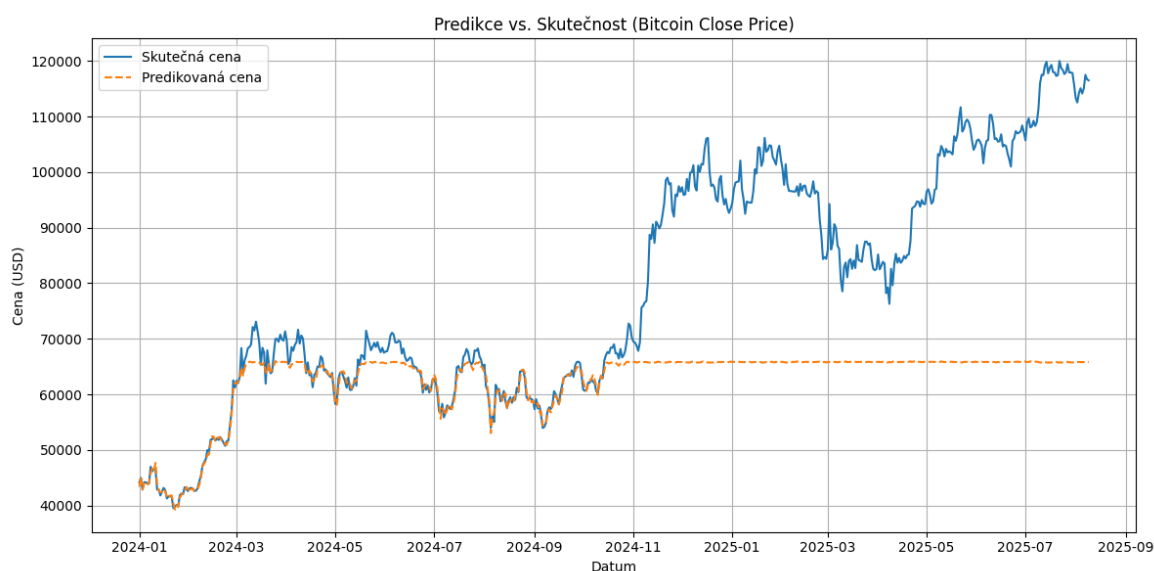
3.7.2 Zhodnocení výsledků

První pokus měl data rozdělena rokem 2022. Model měl dále predikovat cenu až po rok 2025. Model byl schopný kopírovat i menší krátkodobé změny. Bylo dosaženo chyby MAPE 9 %. Ensemble kombinoval LightGBM s RandomForestem a díky tomu měl nejlepší výsledky s hodnotou RMSE okolo 431.

Druhý pokus, který posunul hranici rozdělení dat až na rok 2024, se podle průměrné chyby zhoršil až na MAPE 16 %. Pokud ale porovnáme obrázky 14 a 15, je zřejmé že špatné predikci začínají ve stejném bodě, kdy se více a více opoždíuje. Model také není schopen odhadnout vyšší cenu než zhruba 65 000. Může být způsobeno tím, že v historických datech se takto vysoko cena nikdy nedostala.



Obrázek 14 Predikce Bitcoinu od roku 2022

(zdroj: vlastní)

Obrázek 15 Predikce Bitcoinu od roku 2024

(zdroj: vlastní)

3.8 Experiment 6 – rozpoznávání obrázků

Poslední experiment měl otestovat schopnosti nástroje AutoGluon a jeho možnosti pro trénování modelů v problematice rozpoznávání obrázků. Konkrétně se modely učily na datové sadě obsahující několik druhů jablek. Obrázky byly rozděleny do složek se jmény jednotlivých druhů. Jedná se o, v dnešní době, běžnou potřebu klasifikace vizuálních dat. Především bylo za úkol ukázat, jestli modely trénované pouze na CPU budou ukazovat rozumnou míru přesnosti.

3.8.1 Metodika

- Datová sada: Sada obsahující obrázky šesti různých typů jablek o velikosti 100x100 pixelů (Obrázek 29)
- Časový limit: 10800 s
- Preset: medium
- Typ úlohy: Klasifikace
- Cílová proměnná: Název složky jednotlivých druhů jablek



Obrázek 16 Ukázka obrázku z datové sady

(zdroj: vlastní)

AutoGluon si připravil data a poté vybral neuronovou architekturu vhodnou pro klasifikaci obrázků. Následně určil, aby byl trénován model `TimmAutoModelForImagePrediction`, který má přes 95 milionu parametrů, jak dokazuje obrázek 30. Trénování probíhá po epochách, během kterých probíhá také validace na testovací množině.

	Name	Type	Params	Mode
0	model	TimmAutoModelForImagePrediction	95.7 M	train
1	validation_metric	MulticlassAccuracy	0	train
2	loss_func	CrossEntropyLoss	0	train

Obrázek 17 Model zvolený pro predikci obrázku

(zdroj: vlastní)

Při první validaci bylo dosaženo přesnosti 0,2143 (obrázek 36). Z toho vyplývá, že model ještě nemá naučené žádné vzory a pouze hádá mezi třídami. Pro hluboké modely je to normální chování, při kterém se testují váhy.

Epoch 0, global step 1: 'val_accuracy' reached 0.21429 (best 0.21429)

Obrázek 18 První validace

(zdroj: vlastní)

U druhé validace byla zjištěna přesnost 0,4841 (obrázek 32) a to je více jak dvojnásobné zlepšení. Z toho možné vyvodit schopnost modelu rozpoznávat základní rozdíly mezi jablky jako barva a textura.

Epoch 0, global step 3: 'val_accuracy' reached 0.48413 (best 0.48413)

Obrázek 19 Druhá validace

(zdroj: vlastní)

Po dokončení třetí validace je model schopen správně určit druh jablka ve dvou případech ze tří (obrázek 33). To začíná poukazovat na schopnost rozeznat malé rozdíly mezi podobnými druhy jablek.

Epoch 1, global step 5: 'val_accuracy' reached 0.66667 (best 0.66667)

Obrázek 20 Třetí validace

(zdroj: vlastní)

Další validace už nevykazuje tak razantní nárůst přesnosti jako předchozí. Bylo dosaženo přesnosti téměř 80% (obrázek 34).

Epoch 1, global step 7: 'val_accuracy' reached 0.79365 (best 0.79365)

Obrázek 21 Čtvrtá validace

(zdroj: vlastní)

U předposlední validace je přesnost vyšší než 90% (obrázek 35).

Epoch 2, global step 9: 'val_accuracy' reached 0.91270 (best 0.91270)

Obrázek 22 Pátá validace

(zdroj: vlastní)

Poslední validace dosáhla perfektního skóre 1 (obrázek 36), což znamená správné přiřazení všech tříd v testovací množině.

Epoch 2, global step 11: 'val_accuracy' reached 1.00000 (best 1.00000)

Obrázek 23 Poslední validace

(zdroj: vlastní)

3.8.2 Zhodnocení výsledků

Pro úlohu rozpoznávání ovoce, AutoGluon využil MultimodalPredictor a bez využití GPU, byl schopen model dosáhnout 100% přesnosti. Celý experiment trval okolo 50 minut, poté byl automaticky ukončen. Trénování trvalo 3 epochy mezi kterými je vidět narůst v kvalitě. Z čehože je možné usoudit schopnost sítě se rychle učit základní rozdíly mezi vizuálními třídami. V druhé a zároveň poslední epoše byla přesnost modelu dokonalá, což je působivý výsledek ale je to také důvod k nejistotě. Je možné, že model byl přeučen a nebude schopen přesně rozeznávat nové příklady. Před nasazením by tedy bylo vhodné otestovat model na zcela nové sadě, která obsahuje jiné obrázky jablek.

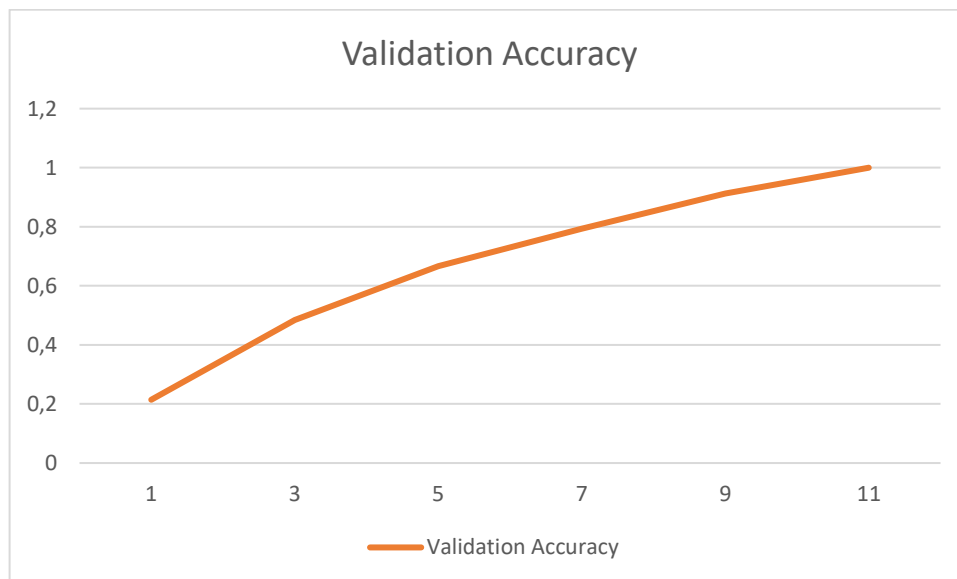
Tento experiment dokazuje, že AutoGluon je vhodným nástrojem pro použití tvorby modelů zaměřené na úlohy typu image classification. Zároveň není za potřebí zasahovat do architektury nebo měnit parametry, což uživatelům šetří čas a práci. Ukazuje jeho silné stránky jako jsou robustní výsledky a jednoduchost použití v praxi.

Tabulka 12 Výsledná data modelu pro rozpoznávání obrázků

Epoch	Step	Validation Accuracy
0	1	0,2143
0	3	0,4841
1	5	0,6667

1	7	0,7936
2	9	0,9127
2	11	1,0000

(zdroj: vlastní)



Obrázek 24 Graf z tabulky 16

(zdroj: vlastní)

ZÁVĚR

Tato práce popsala fungování automatizace strojového učení. Poukázala na to, jaké možnosti využití nabízí nástroj AutoGluon. Tyto možnosti byly prezentovány na provedených experimentech. V teoretické části bylo vysvětleno automatizované strojové učení a byl přiblížen proces vývoje jeho modelů. Dále také vysvětlila možné využití v různých oborech. Byly také vysvětleny techniky, které jsou při AutoML využívány.

Informace popsané v teoretické části byly poté ukázány na experimentech v praktické části. První experiment dokázal, že volí typ úlohy podle sledované typu proměnné. Pokud tedy předem neurčíme jinak, AutoGluon si zvolí sám a může natrénovat modely jiného typu, než bylo očekáváno. Další experiment testoval možnosti změny trénovacího času, kde je z výsledků možno vyvodit, že delší trénink ne vždy dosáhne lepších výsledků.

Velmi hodnotných výsledků dosáhl experiment zkoumající vliv hyperparametrů na jednotlivé modely. Zde výsledky poukazují na důležitost vyvážení kroku, po kterých se modely učí, a počtu iterací učení.

Dalším hodnotných poznatků bylo dosaženo experimentem, který pracoval s historickými cenami Bitcoinu. Z pokusů s těmito daty vyplývá že modely natrénované AutoGluonem jsou schopné docela přesně kopírovat běžný vývoj, ale má problémy při prudkých nárustech hodnot. To poukazuje na limity klasických přístupů při extrapolaci časových řad mimo hodnoty historických vzorů.

Poslední experiment měl skvělý úspěch, kdy bylo dosaženo perfektní přesnosti. Model měl za úkol rozpoznat obrázky ovoce a při testování určil všechny správně. Může nastat obava, tento model byl přeučen, a proto dosáhl tak dobrého výsledku. Jediný způsob, jak tomuto předejít je před nasazením modelu jej otestovat na naprosto nové sadě.

SEZNAM POUŽITÉ LITERATURY

- [1] MITCHELL, Tom. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN 9780070428072.
- [2] BISHOP, Christopher M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. ISBN 9780387310732.
- [3] MURPHY, Kevin P. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012. ISBN 9780262018029.
- [4] HASTIE, Trevor, Robert TIBSHIRANI a Jerome FRIEDMAN. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. vyd. New York: Springer, 2009. ISBN 9780387848570.
- [5] GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. *Deep Learning*. Cambridge, MA: MIT Press, 2016. ISBN 9780262035613.
- [6] HUTTER, Frank, Lars KOTTHOFF a Joaquin VANSCHOREN (eds.). *Automated Machine Learning: Methods, Systems, Challenges*. Cham: Springer, 2019. ISBN 9783030053178.
- [7] CHOLLET, François. *Deep Learning with Python*. 2. vyd. New York: Manning, 2021. ISBN 9781617296864.
- [8] GÉRON, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 3. vyd. Sebastopol: O'Reilly Media, 2022. ISBN 9781098125974.
- [9] ZHANG, Aston, Zachary LIPTON, Mu LI a Alexander SMOLA. *Dive into Deep Learning*. Cambridge: Cambridge University Press, 2020. ISBN 9781108470049.
- [10] JORDAN, Michael I. a Tom M. MITCHELL. Machine learning: Trends, perspectives, and prospects. *Science*. 2015, 349(6245), 255–260. ISSN 0036-8075. DOI: 10.1126/science.aaa8415.
- [11] SHWARTZ-ZIV, Ravid a Amit ARMON. Tabular Data: Deep Learning is Not All You Need. *arXiv preprint*, arXiv:2106.03253, 2021.
- [12] SNOEK, Jasper, Hugo LAROCHELLE a Ryan P. ADAMS. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems (NIPS)*. 2012.
- [13] VANSCHOREN, Joaquin, Jan N. VAN RIJN, Bernd BISCHL a Luis TORGOSO. OpenML: Networked Science in Machine Learning. *ACM SIGKDD Explorations Newsletter*. 2013, 15(2), 49–60. ISSN 1931-0145.
- [14] MASOOD, Adnan. *Automated Machine Learning: Hyperparameter Optimization, Neural Architecture Search, and Algorithm Selection with Cloud Platforms*. Birmingham: Packt Publishing, 2021. ISBN 9781800565524.
- [15] SABHARWAL, Navin a Amit AGRAWAL. *Up and Running Google AutoML and AI Platform: Building Machine Learning and NLP Models Using AutoML and AI Platform for Production Environment*. Delhi: BPB Publications, 2020. ISBN 9789388511921.
- [16] LAT, Joshua Arvin. *Machine Learning Engineering on AWS: Build, scale, and secure machine learning systems and MLOps pipelines in production*. Birmingham: Packt Publishing, 2022. ISBN 9781803247596.
- [17] MCKINNEY, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2. vyd. Sebastopol: O'Reilly Media, 2017. ISBN 9781491957660.
- [18] ERICKSON, Nick, Jonas MUELLER, Alexander MENDES, Steve KINOSHITA, Boris STANCIULESCU aj. AutoGluon-tabular: Robust and Accurate AutoML for Structured Data. *arXiv preprint*, arXiv:2003.06505, 2020.

- [19] ZÖLLER, Marc-André a Marco F. HUBER. Benchmark and Survey of Automated Machine Learning Frameworks. *Journal of Artificial Intelligence Research*. 2019, 70, 409–472. ISSN 1076-9757. DOI: 10.1613/jair.1.11692.
- [20] HE, Xin, Kai ZHAO a Xiaowei CHEN. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*. 2019, 194, 105–129. ISSN 0950-7051. DOI: 10.1016/j.knosys.2020.105129.
- [21] AUTOGLUON TEAM. *AutoGluon Documentation (v1.2.0)* [online]. Amazon Web Services, 2023 [cit. 2025-08-21]. Dostupné z: <https://auto.gluon.ai/1.2.0/index.html>
- [22] AUTOGLUON TEAM. *AutoGluon Documentation (stable)* [online]. Amazon Web Services, 2023 [cit. 2025-08-21]. Dostupné z: <https://auto.gluon.ai/stable/index.html>
- [23] GOOGLE CLOUD. *Vertex AI: Beginner's Guide* [online]. Google, 2023 [cit. 2025-08-21]. Dostupné z: <https://cloud.google.com/vertex-ai/docs/beginner/beginners-guide>
- [24] DATADOG. *Monitoring Azure Machine Learning with Datadog* [online]. DatadogHQ, 2023 [cit. 2025-08-21]. Dostupné z: <https://www.datadoghq.com/dg/monitor/azure-machine-learning/>
- [25] AMAZON WEB SERVICES. *Amazon SageMaker Autopilot* [online]. AWS, 2023 [cit. 2025-08-21]. Dostupné z: <https://aws.amazon.com/sagemaker/ai/autopilot/>
- [1]

SEZNAM OBRÁZKŮ

Obrázek 1 Ukázka datové sady jablek	29
Obrázek 2 Graf z tabulky 3	31
Obrázek 3 Graf z tabulky 4	32
Obrázek 4 Graf z tabulky 5	33
Obrázek 5 Graf z tabulky 6	34
Obrázek 6 Graf z tabulky 7	35
Obrázek 7 Rozptyl predikcí pro limit 15 s	36
Obrázek 8 Rozptyl predikcí pro limit 100 s	37
Obrázek 9 Graf z tabulky 8 porovnání MAE a RMSE metrik	38
Obrázek 10 Graf z tabulky 9 porovnání MAE a RMSE metrik	39
Obrázek 11 Rozptyl predikcí pro třetí kombinaci	41
Obrázek 12 Graf z tabulky 10	42
Obrázek 13 Graf z tabulky 11	44
Obrázek 14 Predikce Bitcoinu od roku 2022	45
Obrázek 15 Predikce Bitcoinu od roku 2024	45
Obrázek 16 Ukázka obrázku z datové sady	46
Obrázek 17 Model zvolený pro predikci obrázku	47
Obrázek 18 První validace	47
Obrázek 19 Druhá validace	47
Obrázek 20 Třetí validace	47
Obrázek 21 Čtvrtá validace	47
Obrázek 22 Pátá validace	48
Obrázek 23 Poslední validace	48
Obrázek 24 Graf z tabulky 16	49

SEZNAM TABULEK

Tabulka 1 Ukázka datové sady pro mobilní data	27
Tabulka 2 Parametry prostředí	29
Tabulka 3 Výsledné hodnoty pro preset medium	30
Tabulka 4 Výsledné hodnoty pro preset good	32
Tabulka 5 Výsledné hodnoty pro preset high	33
Tabulka 6 Výsledné hodnoty pro preset best	34
Tabulka 7 Výsledné hodnoty pro preset experimental.....	35
Tabulka 8 Výsledné metriky pro limit 15 s.....	37
Tabulka 9 Výsledné metriky pro limit 100 s.....	38
Tabulka 10 Porovnání výsledných metrik hyperparametrů	42
Tabulka 11 Výsledná RMSE hodnota pro experiment 4	43
Tabulka 16 Výsledná data modelu pro rozpoznávání obrázků	48

SEZNAM PŘÍLOH

Příloha P I: Dataset *mobile_data* – tabulková data obsahující specifikace mobilních telefonů a jejich prodejní ceny, využita pro klasifikaci a regresi.

Příloha P II: Dataset *bitcoin_history* – časová řada denních cen Bitcoinu (Open, High, Low, Close, Volume), použita pro predikci vývoje ceny.

Příloha P III: Dataset *Fruits* – obrazová data s fotografiemi ovoce, využita pro úlohu klasifikace obrázků.

Všechny datové sady jsou dostupné v repozitáři:
<https://github.com/RadekSusen/autogluon>

PŘÍLOHA P I: NÁZEV PRVNÍ PŘÍLOHY

Obsah první přílohy.