FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Predikce materiálových vlastností vzorků vyráběných procesem válcování

Diplomová práce

Vedoucí práce:
Ing. Luboš Šmídl, Ph.D.

Vypracoval:
Bc. Valentin Papazian

Plzeň 2024

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Valentin PAPAZIAN**
Osobní číslo:	**A22N0096P**
Studijní program:	**N0714A150011 Kybernetika a řídicí technika**
Specializace:	**Umělá inteligence a automatizace**
Téma práce:	**Predikce materiálových vlastností vzorků vyráběných procesem válcování**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování

1. Seznamte se s problematikou procesu válcování kovů, jeho vstupními a procesními parametry majícími vliv na výslednou kvalitu produktu.
2. Analyzujte dosavadní přístupy k řešení dané problematiky.
3. Navrhněte řešení s využitím nástrojů strojového učení, vhodné metody implementujte.
4. Diskutujte dosažené výsledky a jejich přesnost.
5. Popište omezení a navrhněte vylepšení s ohledem na možnost nasazení v průmyslové praxi.

Rozsah diplomové práce: **40-50 stránek A4**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Ing. Luboš Šmídl, Ph.D.**
Katedra kybernetiky

Datum zadání diplomové práce: **2. října 2023**
Termín odevzdání diplomové práce: **20. května 2024**

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Dr. Ing. Vlasta Radová
vedoucí katedry

V Plzni dne 2. října 2023

Poděkování

Rád bych vyjádřil své poděkování panu Ing. Luboši Šmídlovi, Ph.D. za vedení a cennou podporu při vypracování mé diplomové práce. Velké díky také patří panu Ing. Filipu Polákovi za konzultace. Děkuji také panu Ing. Janu Knoblochovi ze společnosti PTSW za rady, které byly klíčové pro praktickou část mé práce. Mé poděkování patří také mé rodině a přátelům za podporu během celého studia.

Prohlášení

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

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

V Plzni dne 17.5.2024

.....
Bc. Valentin Papazian

Abstrakt

Tato diplomová práce se zabývá vývojem modelu pro efektivní predikci síly potřebné k válcování plechů, což je klíčový aspekt v procesu výroby kovových materiálů. Využívá metody strojového učení a provádí rozsáhlou analýzu dat, zkoumá různé algoritmy, včetně lineární regrese, k-Nearest Neighbors (kNN) a stromových metod. Data pro analýzu byla poskytnuta společností PT Solutions Worldwide (PTSW) a obsahují údaje z válcovny hliníku za studena.

Přesné predikce mají za cíl nejen snížit výrobní náklady, ale také zvýšit kvalitu a konzistenci finálních produktů. Práce dále identifikuje potenciální směry pro budoucí výzkum, jako je testování modelů na rozšířených a diverzifikovaných datasetech nebo vytvoření ensemble modelů.

Testování algoritmů strojového učení s různými předzpracováními dat odhalilo, že model kNN s logaritmickým předzpracováním dat je pro tuto specifickou úlohu nejvhodnější, dosahující MAPE 6.35 %.

Tato studie přináší informace o integraci strojového učení do průmyslových procesů a nastiňuje možnosti jejich dalšího vývoje a optimalizace.

Klíčová slova

Strojové učení, datová analýza, předzpracování dat, válcování, prediktivní analýza, učení s učitelem.

Abstract

This thesis focuses on the development of a predictive model for effectively forecasting the force required for rolling sheet metal, a key aspect in the production process of metallic materials. It utilizes machine learning methods and conducts extensive data analysis, examining various algorithms, including linear regression, k-Nearest Neighbors (kNN), and tree-based methods. The data for analysis was provided by PT Solutions Worldwide (PTSW). Accurate predictions aim not only to reduce manufacturing costs but also to enhance the quality and consistency of the final products. The work further identifies potential directions for future research, such as testing models on expanded and diversified datasets or creating ensemble models. Testing machine learning algorithms with various data preprocessing revealed that the kNN model with logarithmic data preprocessing is the most suitable for this specific task, achieving a MAPE of 6.35 %. This study provides insights into the integration of machine learning into industrial processes and outlines possibilities for their further development and optimization.

Key words

Machine learning, data analysis, preprocessing, rolling, predictive analysis, supervised learning.

Obsah

1 Úvod

Tato diplomová práce se zaměřuje na analýzu a predikci síly potřebnou pro válcování plechů, což je zásadní aspekt ve výrobním procesu kovových materiálů. Válcování je technologický proces, při kterém dochází k deformaci materiálu procházejícího mezi dvěma nebo více válci. Přesná predikce síly válců je klíčová pro optimalizaci tohoto procesu, snížení nákladů na výrobu a zajištění kvality konečného produktu.

Cílem této diplomové práce je navrhnout a vyvinout matematický model, který bude schopen předpovídat nutnou sílu válců na základě vstupních parametrů, jako jsou materiál plechu, jeho počáteční tloušťka a požadovaná tloušťka po válcování s tím, aby byla odchylka výstupní tloušťky od požadované co nejmenší. Správná predikce požadované síly povede nejen ke snížení výrobních nákladů, ale také k zajištění vyšší kvality konečných výrobků. Model bude vytvořen s využitím metod strojového učení a statistické analýzy.

V teoretické části práce budou nejprve představeny základní principy válcování. Dále bude představena firma PT Solutions Worldwide (PTSW), která dodala potřebná data a řeší problematiku predikce síly válců. Poté budou diskutovány různé typy předzpracování dat společně s algoritmy strojového učení.

V praktické části práce bude popsán proces přípravy dat pro analýzu, výběr vhodných algoritmů strojového učení, implementace modelu a jeho testování na reálných průmyslových datech. Na závěr budou prezentovány výsledky testování, diskutovány limitace modelu a navrženy možnosti dalšího vývoje.

2 Teoretická část

V této kapitole jsou popsány všechny důležité pojmy a vztahy, které jsou využity v praktické části. První dvě podkapitoly popisují obecně proces válcování a představují společnost PTSW (Process Technology Solutions Worldwide). Následující tři podkapitoly se věnují oblastem předzpracování dat a algoritmům strojového učení. V závěru je podrobně popsán princip fungování Autoencoderu a metody PCA.

2.1 Základy válcování

Válcování představuje technologický proces, který je klíčový pro výrobu a zpracování široké škály materiálů. Jeho hlavní funkcí je deformace materiálu pro dosažení požadovaných změn v jeho tvaru a tloušťce. Materiál, typicky slitina, prochází mezi dvěma paralelně umístěnými válci, které jsou mechanicky poháněny. Tento proces zajišťuje, že materiál je efektivně stlačován a formován tak, aby splňoval specifikované parametry.

Válcováním nedochází pouze k vizuálním a geometrickým změnám materiálu, ale zásadně se mění i jeho vnitřní struktura. Tento proces ovlivňuje mikrostrukturu materiálu, což má přímý dopad na jeho vlastnosti, jako jsou pružnost, pevnost a tvrdost. Tyto změny jsou klíčové pro zajištění funkčních charakteristik finálních výrobků, což umožňuje materiálu vyhovět specifickým nárokům různých průmyslových aplikací.

Válcování tak slouží jako nezbytný krok v průmyslových procesech, jehož cílem je vytvářet výrobky a komponenty s přesnými geometrickými vlastnostmi, což je realizováno změnou původního tvaru a rozměru materiálu prostřednictvím řízeného a strategického aplikování síly. Tento proces má širokou aplikaci ve výrobě a zpracování materiálů a je zodpovědný za produkci širokého spektra průmyslových komponent, včetně plechů, drátů a trubek, a hraje nezastupitelnou roli v sektorech, jako jsou automobilový průmysl, stavitelství, letecký průmysl a mnoho dalších. Válcování je tedy nezbytným a univerzálním procesem v dnešním průmyslovém světě výroby [1].

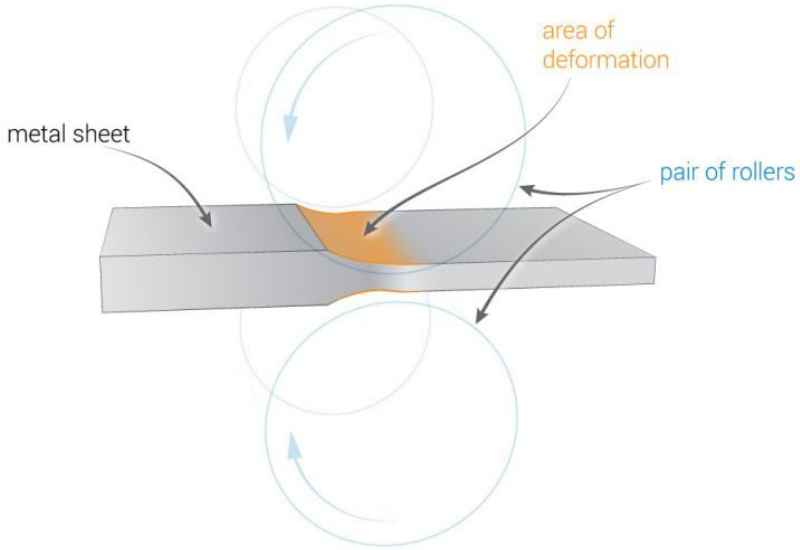
2.1.1 Válcování za studena

Válcování za studena se realizuje při relativně nízkých teplotách, často okolo pokojové, což má za následek zvýšení pevnosti a tvrdosti materiálu, ale za cenu snížení jeho plasticity. Tato technika umožňuje vytvářet výrobky s vysokou přesností a vysoce kvalitním hladkým povrchem, činící ji ideální pro aplikace, kde jsou tyto parametry klíčové jako v automobilovém průmyslu či výrobě nářadí [2].

2.1.2 Válcování za tepla

Na druhé straně, válcování za tepla probíhá za podstatně vyšších teplot, konkrétně nad rekrystalizační teplotou materiálu (většinou 350 °C - 1500 °C), což zajišťuje, že materiál zůstane dostatečně duktilní (duktilita - míra, do jaké se může materiál deformovat před dosažením meze pevnosti) a plastický během celého procesu. I když tato metoda nemůže

nabídnout stejnou úroveň přesnosti a kvality povrchové úpravy jako studené válcování, je mnohdy ekonomičtější (je nutná menší síla k deformaci) a umožňuje efektivní zpracování větších objemů materiálu. Válcování za tepla je tak často využíváno v konstrukčních a stavebních aplikacích, kde jsou kritéria pro estetiku a přesné rozměry méně přísná [3].



Obrázek 1: Válcovací proces [2]

2.2 Proces válcování

Proces válcování zahrnuje sérii strategicky plánovaných fází, které se pohybují od výběru materiálu až po finální výstup a kontrolu kvality, s každou fází přinášející svůj unikátní význam a vyžadující specifickou pozornost.

2.2.1 Výběr materiálu

Prvním krokem je výběr a hodnocení vhodného materiálu, který má požadované vlastnosti, jako jsou pevnost, tvrdost a duktilita. Materiál by měl být vybrán s ohledem na jeho schopnost odolat požadovaným mechanickým zatížením a s ohledem na environmentální faktory, jako je odolnost proti korozi.

2.2.2 Příprava materiálu

Další fáze je příprava materiálu. Tato etapa zahrnuje čištění povrchu tak, aby byly odstraněny nečistoty a nežádoucí rezidua, což je nezbytné pro dosažení hladkého a rovnoměrného výsledku válcování. Dále předehřívání, které je důležité pro zajištění, že materiál bude mít správnou konzistenci a plasticitu potřebnou pro efektivní deformaci.

V případě některých speciálních aplikací se může k materiálu přidávat i legování (přidání směsi látek za účelem vylepšení vlastností), což může zvýšit jeho pevnost, odolnost

nebo jiné požadované vlastnosti. Materiál, často ve formě bloku, svitku nebo plátku, je po těchto předzpracujících krocích připraven k zavedení do samotného procesu válcování, kde je transformován do požadovaného tvaru a rozměru.

2.2.3 Válcování

Samotné válcování začíná zavedením připraveného materiálu do pracovní zóny válce. Následně jsou válcovací válce roztočeny a materiál je veden mezi ně. Aplikovaný tlak a směr sil od válců způsobuje deformaci materiálu, který je tak formován do požadované tloušťky a tvaru. Dynamika a parametry průchodu, jako jsou rychlost válcování a rozestupy válců, musí být pečlivě regulovány, aby byly splněny specifikace výsledného produktu. V některých případech je materiál veden postupně několika válcovacími hlavicemi za sebou, což vyžaduje vysokou přesnost změny tloušťky materiálu po každém průchodu. Většinou dojde k válcování za tepla a poté k válcování za studena. Nicméně oba tyto procesy lze provést vícekrát.

Materiál může být opakovaně válcován za tepla ve více pasážích, což umožňuje postupné tvarování a řízení struktury materiálu. Více etapové procesy pomáhají optimalizovat vlastnosti materiálu, jako jsou pevnost, tvrdost a duktilita, a také zlepšují homogenitu struktury.

Po procesu válcování za tepla a následném chlazení může být materiál dále zpracováván válcováním za studena. Studené válcování může být také provedeno v několika etapách, přičemž mezi jednotlivými pasážemi může být materiál žíhán, aby se obnovila jeho duktilita a snížila vzniklá vnitřní napětí.

2.2.4 Normalizace a chlazení

Dohřívání a chlazení v procesu válcování významně ovlivňují finální vlastnosti zpracovávaného materiálu. Fáze normalizace, často nezbytná pro válcování za tepla, vyžaduje dohřívání materiálu na specifickou teplotu, která je následně udržována po určitý časový interval, aby bylo dosaženo požadované homogenity a zlepšení mechanických vlastností.

Následné pomalé chlazení pak poskytuje kontrolovanou cestu k dosažení specifikované mikrostruktury a optimalizaci mechanických vlastností. Na druhou stranu, rychlost chlazení je rovněž faktor, jenž může být záměrně modulován, tak aby byly získány specifické a požadované vlastnosti materiálu.

Tato chladicí fáze může být realizována prostřednictvím různých technik a médií, včetně vzduchu, oleje či vody, každá s vlastními charakteristickými výhodami a omezeními, aby byla splněna specifika konkrétního výrobního procesu a jeho výsledného produktu.

Tento celý proces od dohřívání až po kontrolované chlazení je tedy zásadní pro aspekty jako je uniformita (rovnoměrnost vlastností a struktury materiálu napříč jeho objemem), kvalita a funkčnost výsledných válcovaných materiálů.

2.2.5 Finální úpravy

Po těchto procesech materiál prochází fázemi tvarování, kde může být dále upravován prostřednictvím dalších válcovacích operací, jako jsou například zakřivování, stříhání či profilování, aby tak odpovídal konečným specifikacím. V tomto stádiu může také dojít k dodatečným tepelným a povrchovým úpravám, jako jsou žíhání (zahřátí materiálu a následné ochlazení na volném vzduchu), chlazení nebo povlakování, což vylepšuje vlastnosti a vizuální kvalitu výrobku.

Povrchové úpravy slouží k vylepšení estetických a funkčních vlastností výrobku. Aplikace různých povlaků může nabídnout lepší odolnost proti korozi nebo zvýšit odolnost proti opotřebení, leštění může vylepšit vizuální estetiku materiálu a další úpravy povrchu mohou být prováděny tak, aby byly splněny konkrétní požadavky týkající se například tření, adheze nebo jiných parametrů relevantních pro výslednou aplikaci materiálu.

2.3 PT Solutions Worldwide (PTSW)

Společnost PTSW se specializuje na poskytování komplexních automatizačních řešení pro průmyslový sektor, zejména pro hutní průmysl a energetiku. Firma existuje už více než 20 let. Její řešení jsou navržena tak, aby byla provozována v reálném čase a reprezentovala špičku v oboru, čímž klientům pomáhají splnit jak procesní, tak obchodní požadavky [4].

Kompletní služby nabízené PTSW zahrnují:

- Návrh procesu: Vytváření efektivních a optimalizovaných procesních plánů pro klienty.
- Základní a detailní inženýring: Rozpracování technických specifikací a podrobností pro implementaci automatizačních systémů.
- Integrace aplikací: Sestavení a integrace softwaru a hardwaru pro hladký chod automatizovaných systémů.
- Rozsáhlé školení: Poskytování školení pro zaměstnance klienta, aby mohli efektivně využívat nově implementované systémy.
- Uvedení do provozu: Zajištění, že všechny systémy a komponenty jsou správně nainstalovány, otestovány a plně funkční.
- Podpora po uvedení do provozu: Nabídka pokračující podpory a údržby zařízení a systémů po jejich spuštění.

3 Strojové učení

Strojové učení je oblast umělé inteligence, která umožňuje softwarovým systémům zlepšovat své výkony na základě zkušeností, tedy na základě analýzy dat. Jedná se o techniku, která se soustředí na vývoj algoritmů, jež se mohou učit z dat a dělat predikce nebo rozhodnutí bez explicitního programování k těmto úkolům.

Úspěch strojového učení silně závisí na kvalitě a množství dostupných dat. Špatná kvalita dat nebo jejich nedostatek může vést k nepřesným nebo zkresleným modelům [5].

Základní pojmy

- Data: Základem pro strojové učení jsou data. Ty mohou být ve formě obrázku, textu, zvuku nebo číselných hodnot a jsou nezbytná pro trénink modelů.
- Modely: Definují vztah mezi vstupy a očekávanými výstupy. Modely jsou 'trénovány' na základě historických dat s cílem naučit se předpovídat výstupy na nových datech.
- Trénink: Proces, při kterém se model 'učí' z dat. Během tréninku se model snaží minimalizovat chyby mezi svými predikcemi a skutečnými výsledky a postupně se přizpůsobuje, aby poskytoval přesnější predikce.
- Evaluace: Po tréninku je model vyhodnocen pomocí testovací sady dat, která nebyla použita při tréninku. To umožňuje ověřit, jak dobře model funguje a jak přesné jsou jeho predikce v praxi.

3.1 Předzpracování dat

Předzpracování dat je zásadním krokem ve všech projektech strojového učení, který má klíčový vliv na úspěch a efektivitu výsledných prediktivních modelů. Hrubá data neboli raw data, jsou často neúplná, zašuměná a obsahují irelevantní nebo chybné informace, které mohou významně zkreslit výsledky modelování. Proto je nezbytné provádět důkladné předzpracování dat, aby byla převedena do formátu vhodného pro strojové učení. Tato kapitola poskytuje detailní přehled o technikách a metodách předzpracování dat, včetně normalizace, redukce dimenzí, čištění dat a rozdělení datové sady. Každý z těchto kroků pomáhá zlepšit kvalitu a strukturu datové sady, což je nezbytné pro efektivní trénování a validaci modelů strojového učení [6].

3.1.1 Čištění dat

Čištění dat zahrnuje identifikaci a opravu chyb a inkonzistencí v datových sadách. Tento proces může zahrnovat odstranění duplicitních záznamů, opravu chybných datových typů, vyplnění chybějících hodnot, nebo odstranění 'outliers' (odlehlých hodnot), které mohou zkreslovat analýzu.

3.1.2 Zmenšení dimenze vstupu

Dimenzionalita dat je často snížena za účelem zjednodušení modelů a snížení výpočetní náročnosti. Techniky jako výběr příznaků, extrakce příznaků a redukce dimenze jsou běžně využívány k tomu, aby byly odstraněny nevýznamné nebo redundantní proměnné z datové sady. Mezi základní techniky patří **PCA** a **LDA** [7].

3.1.3 Normalizace dat

Normalizace dat je proces, který upravuje rozsah hodnot atributů v datové sadě tak, aby byly lépe srovnatelné a aby na ně bylo možné aplikovat statistické metody. Metody normalizace zahrnují škálování podle minima a maxima, standardizaci (odstranění průměru a škálování na jednotkovou varianci) a normalizaci podle normy.

Normalizace dat je kritickým procesem ve strojovém učení, který se doporučuje aplikovat v případech, kdy proměnné ve datové sadě mají různý rozsah, ale nevykazují normální rozdělení nebo obsahují odlehlé hodnoty (outliers). Tento proces je zaměřen na úpravu hodnot číselných sloupců tak, aby byly transformovány na společné měřítko bez zkreslení rozdílů v rozsazích hodnot nebo ztráty informace. Jednotné měřítko vstupních veličin je klíčové pro zvýšení efektivity a přesnosti algoritmů strojového učení, tím je zajištěno, že žádný příznak není ve výsledném modelu podceněn nebo přeceňován na základě jeho původního měřítka.

Min-max normalizace (Rescaling)

Tato metoda škáluje data do rozsahu $\langle 0, 1 \rangle$ podle vzorce:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{1}$$

kde x představuje původní hodnoty sloupce, které chceme normalizovat. Metoda je obzvláště užitečná, když potřebujeme zachovat přesné rozdíly v hodnotách, jako je to u algoritmů založených na vzdálenostech.

Z-score normalizace (Standardizace)

Tato technika transformuje data tak, aby měla normální rozdělení se střední hodnotou 0 a směrodatnou odchylkou 1. Používá se vzorec:

$$z = \frac{x - \mu}{\sigma}, \tag{2}$$

kde x reprezentuje hodnoty původního sloupce, μ je střední hodnota a σ je směrodatná odchylka. Tato metoda je vhodná pro algoritmy, které předpokládají, že data jsou normálně rozložena, jako jsou například metody založené na pravděpodobnostních přístupech.

Yeo-Johnson transformace

Yeo-Johnson transformace je statistická metoda používaná pro stabilizaci rozptylu a přiblížení rozdělení dat k normálnímu rozdělení. Tato transformace je rozšířením Box-Cox

transformace a umožňuje transformovat i záporné hodnoty, což Box-Cox transformace nedovoluje [8].

$$y(\lambda) = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda} & \text{pro } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{pro } \lambda = 0, y \geq 0 \\ \frac{-[-(y+1)^{2-\lambda}-1]}{2-\lambda} & \text{pro } \lambda \neq 2, y < 0 \\ -\log(-(y+1)) & \text{pro } \lambda = 2, y < 0 \end{cases} \tag{3}$$

3.1.4 Výběr příznaků

Výběr příznaků je proces vybírání nejrelevantnějších proměnných z datové sady pro použití v modelování. Cílem je zlepšit výkon modelu, zvýšit přesnost a zabránit přetrénování. Výběr příznaků může být realizován pomocí různých statistických, informačních a algoritmických metod.

3.1.5 Feature engineering

Feature engineering je proces, který se zaměřuje na tvorbu nových příznaků z původních dat. To může být užitečné pro učení s učitelem i bez učitele. Cílem je zrychlit a zjednodušit transformaci dat a zároveň zlepšit přesnost modelu. V rámci tohoto procesu se uplatňují různé metody jako:

Polynomial Features, které rozšiřují původní prostor příznaků o nové proměnné získané kombinací existujících. Tyto nové příznaky jsou tvořeny pomocí součinů a mocnin původních proměnných.

Vstupní parametr 'degree' (řád) určuje, jak vysokou mocninu je možné pro generování nových příznaků použít. Například při příznakovém prostoru s proměnnými x a y a hodnotě parametru degree nastavené na 1 by vznikl jeden nový příznak: $[x \cdot y]$. Pokud by byla hodnota degree zvýšena na 2, vznikly by tři nové příznaky: $[x \cdot y, x^2 \cdot y, x \cdot y^2]$.

Tímto způsobem Polynomial Features umožňuje modelům, zejména lineárním, zachytit složitější vzory a interakce mezi proměnnými, což může vést k lepší predikci a interpretaci dat. Rozšíření prostoru příznaků však může také zvýšit riziko přetrénování, zejména u malých datových sad, což vyžaduje pečlivou validaci a případné použití technik pro regulaci modelu.

Další metody mohou třeba být:

Log transformace: U této metody je nezbytné zajistit, že všechny vstupní hodnoty jsou kladné a nenulové, aby bylo možné výpočet provést. Logaritmická transformace je užitečná pro stabilizaci rozptylu a přiblížení distribuce dat k normálnímu rozdělení [9].

Inverzní transformace: Tato metoda vyžaduje, aby žádné z vstupních parametrů nebyly nulové. Inverzní transformace může být účinná pro snížení efektu velmi vysokých hodnot.

3.1.6 Rozdělení dat

Rozdělení dat na trénovací, testovací a validační je klíčové pro úspěšný vývoj strojových učících modelů. Trénovací data umožňují modelu učit se detekovat vzory a souvislosti, zatímco testovací data, která by měla být nezávislá na trénovacích, ověřují jeho schopnost generalizace a pomáhají identifikovat případné přeučení modelu. Validační data se využívají k jemnému ladění parametrů během trénování, což zabraňuje přeučení a zajišťuje, že model si zachovává schopnost generalizace. Ideální rozdělení dat se může lišit pro každou úlohu, běžně je v poměru 80:10:10 pro trénovací, testovací a validační část [10].

4 Algoritmy strojového učení

Algoritmy strojového učení využívají přesně definované sekvence instrukcí k prozkoumávání a interpretaci složitých datových sad. Tyto algoritmy jsou základem pro vytváření modelů, které identifikují vzory použitelné pro kategorizaci dat nebo predikci budoucích výsledků. V závislosti na specifikách dat není vždy zřejmé, který algoritmus bude fungovat nejlépe, což vyžaduje experimentování s různými algoritmy, aby se dosáhlo optimálních výsledků. Většina algoritmů má nastavitelné parametry, známé jako hyperparametry, které mohou zásadně ovlivnit jejich výkonnost. Optimalizace hyperparametrů je proto klíčová. Tento proces zahrnuje hledání nejlepších hodnot pro tyto parametry, což může výrazně zlepšit výkon algoritmu. K automatizaci tohoto procesu se běžně používají techniky jako *grid search* a *random search*, zatímco pokročilejší metody zahrnují *bayesovskou optimalizaci* a *genetické algoritmy* [11]. Algoritmy strojového učení lze rozdělit podle:

Způsobu trénování

- Učení s učitelem (Supervised Learning).
- Učení bez učitele (Unsupervised Learning).
- Zpětnovazební učení (Reinforcement Learning).

Potřebného výstupu

- Regrese.
- Klasifikace.
- Shlukování (clustering).

4.1 Učení s učitelem

Učení s učitelem je základní a nejrozšířenější formou strojového učení, kde cílem je naučit se mapovat vstupy na výstupy na základě předem označených trénovacích dat. Tato metoda se zaměřuje na konstrukci prediktivních modelů, které dokážou předvídat výsledky pro neviděná data s co nejvyšší přesností [12].

4.1.1 Základní principy

Učení s učitelem využívá dataset obsahující příklady vstupních dat a odpovídajících výstupů (cílové hodnoty nebo labely). Algoritmus se snaží vyvinout funkci, která co nejpřesněji mapuje vstupy na výstupy. Tento proces zahrnuje dvě hlavní fáze:

Trénování
Algoritmus se učí rozpoznávat vzorce na základě trénovací sady, která obsahuje vstupní data a správné odpovědi.

Testování
Algoritmus je ověřen na testovací sadě, kde musí předpovědět výstupy na základě dříve neviděných vstupů.

4.1.2 Algoritmy učení s učitelem

Několik běžných algoritmů využívaných v učení s učitelem zahrnuje:

- **Lineární regrese**: Používá se pro predikci spojitých hodnot.
- **Logistická regrese**: Vhodná pro binární klasifikační problémy.
- **Rozhodovací stromy a náhodné lesy**: Efektivní pro klasifikaci i regresi, výhodou je snadná interpretovatelnost výsledků.
- **Support Vector Machines (SVM)**: Silný klasifikační nástroj pro lineární i nelineární data.
- **Neuronové sítě**: Velmi flexibilní modely, vhodné pro komplexní vztahy mezi vstupy a výstupy.

4.1.3 Evaluace modelu

Evaluace výkonu modelu učení s učitelem je klíčová pro zajištění jeho spolehlivosti a generalizovatelnosti.

Nejčastěji používané metriky zahrnují:

- **Přesnost (Accuracy)**: Procento správně klasifikovaných příkladů.
- **MAE, RMSE a MAPE**: Metriky pro měření chyby v regresních úlohách.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \tag{4}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{6}$$

  kde y_i je skutečná hodnota, $\hat{y}_i$ je předpovězená hodnota a n je počet pozorování.

- **F1 skóre**: Jedná se o harmonický průměr přesnosti (Precision) a úplnosti (Recall).

4.2 Učení bez učitele

Učení bez učitele je druh strojového učení, kde jsou modely trénovány na datech, která neobsahují předem označené odpovědi nebo labely. Cílem je identifikovat skryté vzory nebo struktury v datech bez jakéhokoli zásahu nebo vodítek zvenčí [13].

4.2.1 Základní principy

Učení bez učitele se snaží analyzovat a interpretovat data bez jakýchkoli předchozích znalostí o výstupech. Algoritmy se zaměřují na rozpoznávání struktur, seskupování podobných příkladů a detekci anomálií v datech.

Hlavní cíle jsou:

- **Detekce shluků**: Rozdělení datové sady na skupiny (shluky) podobných instancí.
- **Redukce dimenzionality**: Transformace dat tak, aby byla zachována pouze nejrelevantnější informace.
- **Asociační pravidla**: Identifikace pravidel, která popisují silné vztahy mezi proměnnými v datech.

4.2.2 Algoritmy učení bez učitele

Několik populárních algoritmů využívaných v učení bez učitele zahrnuje:

- **K-means clustering**: Metoda pro shlukování, která rozděluje data do K předem definovaných shluků.
- **Autoencoder**: Převádí data do latentního prostoru a zpět s co nejmenším rozdílem původních a transformovaných dat.
- **Hierarchické shlukování**: Postupné skládání clusterů do stromové struktury na základě jejich blízkosti nebo vzdálenosti.
- **Analýza hlavních komponent (PCA)**: Technika pro redukci dimenzí, která transformuje vysokodimenzionální data do nižších dimenzí s maximální variací.

4.2.3 Evaluace modelu

Evaluace modelů strojového učení bez učitele obvykle závisí na analýze vzdáleností mezi shluky a rozložením dat uvnitř těchto shluků.

Základní metriky jsou:

- **Silhouette skóre**: Metrika, která měří, jak dobře jsou data oddělena do shluků.
- **Davies-Bouldin index**: Hodnotí průměrnou 'podobnost' mezi shluky, kde nižší hodnoty indikují lepší separaci.

4.3 Zpětnovazební učení (Reinforcement learning)

Jedná se o typ strojového učení, kde se agent učí provádět akce ve svém prostředí tak, aby maximalizoval kumulativní odměnu. Tento přístup se liší od učení s učitelem a učení bez učitele tím, že se zaměřuje na vývoj strategie agenta interagujícího s dynamickým prostředím, na rozdíl od predikce pevně definovaného výstupu nebo identifikace vzorů ve statických datech [14].

4.3.1 Základní principy

Zpětnovazební učení se opírá o koncepci agenta, který se rozhoduje na základě stavu prostředí, v němž se nachází. Agent provádí akce a prostředí reaguje na tyto akce změnou stavu a přidělením odměn nebo trestů. Cílem agenta je naučit se strategii zvanou politika, která maximalizuje jeho kumulativní odměny během času.

Klíčové komponenty:

- **Agent**: Rozhodující entita, která vykonává akce.
- **Prostředí**: Svět, ve kterém agent operuje a odkud získává stavové informace a odměny.
- **Politika**: Strategie, podle které agent vybírá akce založené na stavu prostředí.
- **Funkce odměny**: Odhaduje očekávanou kumulativní odměnu z daného stavu nebo stavu a akce.
- **Model prostředí**: Model, který agent používá k předpovědi, jak prostředí reaguje na akce.

4.3.2 Algoritmy zpětnovazebního učení

Existuje mnoho algoritmů zpětnovazebního učení od jednoduchých, jako je Q-learning a SARSA [15], po složitější jako hluboké učení s posilou (např. Deep Q-Networks - DQN), které integrují hluboké neuronové sítě pro schopnost zvládat velké a složité stavy [16].

4.3.3 Evaluace modelu

Evaluace modelů učení s posilou se obvykle provádí na základě efektivity a robustnosti naučené politiky. Hlavním ukazatelem úspěchu je celková odměna, kterou agent získává během definovaného časového úseku, a schopnost adaptace na nové, neviděné stavy prostředí.

4.4 Algoritmy použité v praktické části

Výběr vhodných trénovacích algoritmů je důležitou částí pro vytvoření kvalitního prediktivního modelu. Jelikož neexistuje jeden univerzálně nejlepší algoritmus, je potřeba vyzkoušet více různých algoritmů. Následně jsou popsány algoritmy použité v praktické části.

4.4.1 Lineární regrese

Lineární regrese představuje jednu z nejzákladnějších a nejčastěji používaných metod pro predikci a analýzu vztahů mezi proměnnými. V jádru tohoto přístupu je cíl minimalizovat rozdíly mezi pozorovanými hodnotami závislé proměnné a hodnotami modelovanými na základě lineární kombinace nezávislých proměnných. Matematicky je tento úkol realizován minimalizací součtu kvadrátů odchylek mezi skutečnými výsledky a výsledky predikovanými, což se běžně označuje jako metoda nejmenších čtverců.

Základní matematický model lineární regrese lze formulovat jako přeurčenou soustavu lineárních rovnic:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \tag{7}$$

kde $\mathbf{A}$ představuje matici funkčních hodnot nezávislých proměnných, $\mathbf{x}$ je vektor neznámých koeficientů (vah), které je potřeba určit, a $\mathbf{b}$ reprezentuje vektor pozorovaných hodnot závislé proměnné.

Výsledný vztah pro výpočet vektoru vah $\mathbf{x}$ získáváme derivací kvadratické chybové funkce a následným určením podmínky pro její minimum:

$$\mathbf{x} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}, \tag{8}$$

zde $\mathbf{A}^T$ označuje transponovanou matici $\mathbf{A}$ a součin $\mathbf{A}^T \cdot \mathbf{A}$ je symetrická matice, jejíž inverze umožňuje vypočítat koeficienty tak, aby byla minimalizována chybová funkce. Výraz $(\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T$ je takzvaná pseudoinverzní matice $\mathbf{A}$, který poskytuje řešení nejmenších čtverců pro přeurčené systémy rovnic.

Výhody

- Rychlost.
- Interpretovatelnost.

Nevýhody

- Předpoklad lineárních vztahů.
- Citlivost na 'outliers'.

4.4.2 k-Nearest Neighbors (kNN)

k-Nearest Neighbors (kNN) je jednoduchý a intuitivní algoritmus pro klasifikaci a regresi, který přiřazuje hodnotu nebo třídu neznámého bodu na základě hodnot nebo tříd jeho nejbližších sousedů v příznakovém prostoru.

Klíčové vlastnosti

- Neparametrický model: kNN nečiní žádné předpoklady o tvaru rozdělení dat.
- Lazy learning: kNN, typicky označovaný jako 'lazy learner', nevytváří předem žádný tréninkový model. Veškeré výpočty a vyhodnocení modelu se provádějí až v momentě, kdy je třeba udělat predikci.

Výhody

- Snadná interpretace.
- Flexibilita: kNN může být použit jak pro klasifikaci, tak pro regresi.

Nevýhody

- Výpočetně náročný: Vyžaduje uložení celého tréninkového souboru a může být pomalý při dotazování na velkých datových sadách.
- Citlivost na dimenzionalitu: Výkon kNN může klesat v případech, kdy má datový soubor vysokou dimenzionalitu (tzv. 'curse of dimensionality').
- Náchylnost na šum.

4.4.3 Stromové metody

Stromové algoritmy jsou populární nástroje v oblasti strojového učení díky jejich interpretovatelnosti, flexibilitě a schopnosti pracovat s různými typy dat. Používají se pro klasifikační i regresní úlohy. Rozhodovací strom se skládá z uzlů (**nodes**) představujících rozhodnutí na základě specifických vlastností, větví (**branches**) reprezentujících možné výsledky a listových uzlů (**leaf nodes**) označujících předpovězené hodnoty nebo třídy.

Rozhodovací stromy mají několik výhod: zvládají jak kategorická, tak numerická data, vyžadují minimální předzpracování dat a jsou robustní vůči odlehlým hodnotám a chybějícím údajům. Na druhou stranu mohou být náchylné k přeučení a nestabilitě, zvláště při práci s šumovými daty. Také poskytují diskrétní, skokové předpovědi, což může být pro některé aplikace nevhodné [17].

Následně je představeno pět metod rozhodovacích stromů: rozhodovací stromy (Decision Trees), náhodné lesy (Random Forest), Extra Trees, Gradient Boosting a XGBoost [18].

Rozhodovací stromy (Decision Trees)

Rozhodovací stromy jsou základní formou stromových modelů. Tyto modely pracují na principu rozdělení dat na podmnožiny pomocí pravidel rozhodování na základě hodnot atributů.

Klíčové vlastnosti:

- Snadná interpretace a vizualizace.
- Schopnost zvládat jak číselná, tak kategorická data.
- Náchylnost k přeučení (overfitting), což může být zmírněno prořezáváním (pruning).

Výhody:

- Intuitivní pochopení a interpretace.
- Rychlé a efektivní trénování na malých až středně velkých datech.

Nevýhody:

- Tendence k přeučení.
- Méně přesné v porovnání s komplexnějšími modely.

Náhodné lesy (Random Forest)

Random Forest je ensemble metoda, která kombinuje několik rozhodovacích stromů, aby zlepšila prediktivní výkon a robustnost modelu.

Klíčové vlastnosti:

- Používá náhodné podmnožiny dat a atributů pro trénování jednotlivých stromů.
- Výsledná predikce je kombinací predikcí jednotlivých stromů (např. průměr nebo hlasování).

Výhody:

- Snížené riziko přeučení díky agregaci výsledků z více stromů.
- Vysoká přesnost a robustnost proti šumu v datech.

Nevýhody:

- Ztráta interpretovatelnosti ve srovnání s jednotlivými rozhodovacími stromy.
- Vyšší výpočetní náročnost.

Extra Trees (Extremely Randomized Trees)

Extra Trees je varianta náhodných lesů, která zavádí další náhodnost do procesu trénování stromů.

Klíčové vlastnosti:

- Náhodná volba prahových hodnot pro rozdělení dat, nejen náhodný výběr atributů.
- Vylepšená diverzita stromů, což může vést k lepší generalizaci.

Výhody:

- Rychlejší trénování ve srovnání s Random Forest.
- Dobrá přesnost a robustnost.

Nevýhody:

- Ještě méně interpretovatelný než Random Forest.
- Může trpět stejnými problémy s výpočetní náročností jako Random Forest.

Gradient Boosting

Gradient Boosting je metoda založená na sekvenčním trénování stromů, kde se každý nový strom snaží opravit chyby předchozích stromů.

Klíčové vlastnosti

- Postupné zlepšování modelu přidáváním stromů, které se zaměřují na rezidua chyb.
- Vysoká přesnost při vhodné parametrizaci.

Výhody

- Velmi vysoká přesnost na složitých úlohách.
- Flexibilita díky možnosti použití různých funkcí ztráty.

Nevýhody

- Vyšší riziko přeučení, pokud není správně regulováno.
- Vysoké nároky na tréninkový čas a paměť.

XGBoost (Extreme Gradient Boosting)

XGBoost je optimalizovaná verze Gradient Boosting, která je navržena pro vysokou efektivitu a rychlost.

Klíčové vlastnosti

- Implementace regularizace pro lepší generalizaci.
- Podpora paralelního zpracování a distributed computing.

Výhody

- Vynikající výkon na soutěžích v oblasti strojového učení.
- Rychlejší a efektivnější trénink ve srovnání s tradičním Gradient Boosting.

Nevýhody

- Vyšší komplexnost v porovnání s jinými stromovými modely.
- Potřeba ladění většího počtu hyperparametrů.

4.5 Multilayer perceptron (MLP)

Vícevrstvé perceptrony (MLP) jsou základním typem umělých neuronových sítí, které patří do kategorie dopředných sítí. Tyto sítě jsou založeny na struktuře spojené s několika vrstvami neuronů, kde každý neuron ve vrstvě je plně propojen s neurony ve vrstvě následující. V MLP signály putují od vstupních neuronů skrze jednu nebo více skrytých vrstev až do výstupní vrstvy [19].

4.5.1 Definice neuronu

Neuron je základní stavební jednotkou v neuronových sítích. Každý neuron simuluje chování biologických neuronů a funguje jako výpočetní jednotka, která přijímá vstupy, aplikuje na ně váhy a bias a následně generuje výstup použitím aktivační funkce.

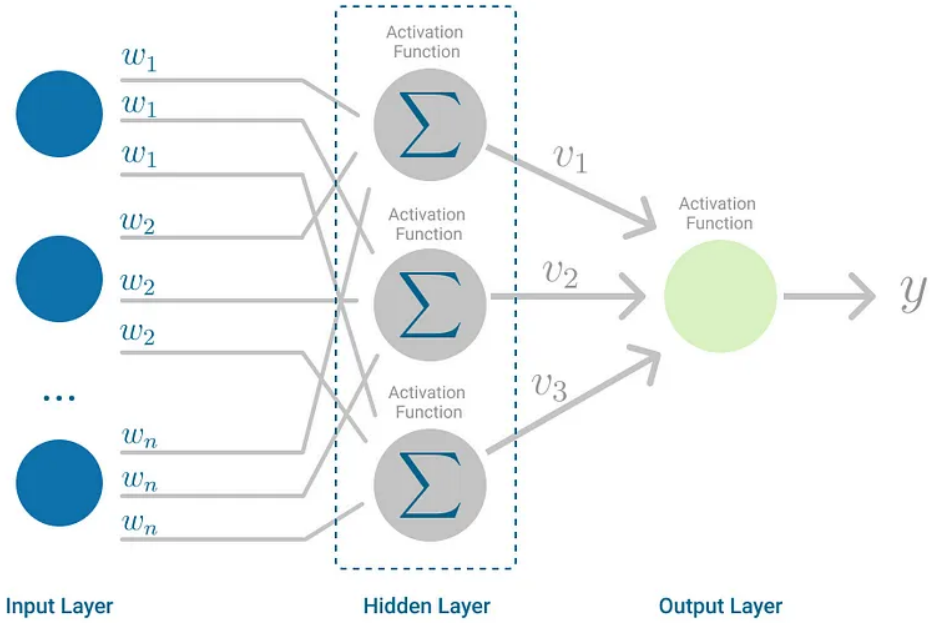
Matematicky je model neuronu ve vícevrstvém perceptronu (MLP) reprezentován následovně:

$$z_j^{(l)} = \sum_{i=1}^{n} w_{ji}^{(l)} x_i + b_j^{(l)}, \tag{9}$$

$$a_j^{(l)} = \sigma(z_j^{(l)}). \tag{10}$$

Ve vztahu 9 označují x_i vstupy do neuronu, $w_{ji}^{(l)}$ představují váhy přiřazené těmto vstupům, $b_j^{(l)}$ označuje bias a σ je aktivační funkce.

Vstup $z_j^{(l)}$ je lineární kombinace vstupů a vah a $a_j^{(l)}$ je výstup neuronu po aplikaci aktivační funkce.



Obrázek 2: Multilayer perceptron [19]

MLP se skládá z několika klíčových komponent:

- **Vstupní vrstva**: Tato vrstva přijímá vstupní data. Počet neuronů v této vrstvě odpovídá počtu vstupních proměnných.
- **Skryté vrstvy**: MLP obsahuje alespoň jednu skrytou vrstvu, ale může jich být více. Skryté vrstvy transformují vstupní signály pomocí vážených lineárních kombinací následovaných nelineárními aktivačními funkcemi.
- **Výstupní vrstva**: Počet neuronů v této vrstvě závisí na specifické úloze, například pro binární klasifikaci je obvykle jeden neuron, zatímco pro více třídní klasifikaci odpovídá počet neuronů počtu tříd.

4.5.2 Aktivační funkce

Aktivační funkce jsou klíčové pro zavedení nelinearity do sítě, což umožňuje MLP modelovat komplexnější funkce.

Některé běžně používané aktivační funkce zahrnují:

- **ReLU (Rectified Linear Unit)**: Rychlá a efektivní, často používaná ve skrytých vrstvách.
- **Sigmoid**: Tradičně používaná pro výstupní vrstvu v binární klasifikaci.

- **Softmax**: Používá se na výstupní vrstvě pro více třídní klasifikaci, protože generuje pravděpodobnosti tříd.

4.5.3 Trénink MLP

MLP se trénuje pomocí algoritmu zvaného zpětné šíření chyby (backpropagation) ve spojení s optimalizačním algoritmem, jako je gradientní sestup (gradient descent).

Tento proces zahrnuje několik kroků:

- **Forward Pass**: Výpočet výstupů sítě a chyby vzhledem k očekávanému výstupu.
- **Backward Pass**: Výpočet gradientů funkce chyby vzhledem k vahám sítě, začínající od výstupní vrstvy a postupující zpět k vstupní vrstvě.
- **Aktualizace vah**: Váhy jsou aktualizovány tak, aby se minimalizovala chyba sítě.

4.5.4 Použití MLP

MLP jsou široce používány pro různé úlohy strojového učení včetně:

- **Klasifikace**: Od binární klasifikace po více třídní klasifikaci.
- **Regrese**: Modelování spojitých výstupních hodnot.
- **Předzpracování dat**: Můžou být použity pro snížení rozměrů nebo extrakci příznaků.

MLP mohou být velmi účinné pro mnoho úloh, ale jejich efektivita závisí na správné architektuře včetně počtu vrstev, počtu neuronů ve vrstvách a volbě aktivačních funkcí. Obecně platí, že MLP jsou silným základem pro hluboké učení (deep learning), ale mohou trpět některými omezeními, jako je přeučení a výpočetní náročnost, zejména při práci s velmi velkými datovými soubory nebo vysokou dimenzionalitou vstupů.

4.6 Autoencoder

Autoencoder je typ umělé neuronové sítě používaný pro efektivní kódování dat. Jeho hlavním cílem je naučit se reprezentovat data v komprimované formě, což je často využíváno pro redukci dimenzí dat nebo pro odstraňování šumu.

Autoencoder se skládá z několika klíčových komponent:

- **Vstupní vrstva:** Přijímá vstupní data. Počet neuronů ve vstupní vrstvě odpovídá počtu vstupních proměnných.
- **Kódovací část (Encoder):** Skládá se z jedné nebo více skrytých vrstev, které transformují vstupní data do latentního prostoru. Proces transformace zahrnuje vážené lineární kombinace následované nelineárními aktivačními funkcemi.
- **Latentní prostor:** Komprimovaná reprezentace vstupních dat. Počet neuronů v této vrstvě je menší než ve vstupní vrstvě, což umožňuje redukci rozměrů.
- **Dekódovací část (Decoder):** Skládá se z jedné nebo více skrytých vrstev, které rekonstruují data z latentního prostoru zpět do původního prostoru.
- **Výstupní vrstva:** Generuje rekonstrukci vstupních dat. Počet neuronů ve výstupní vrstvě odpovídá počtu neuronů ve vstupní vrstvě.

4.6.1 Aktivační funkce

Aktivační funkce jsou klíčové pro zavedení nelinearity do sítě, což umožňuje autoenceroderu modelovat komplexnější vztahy v datech. Některé běžně používané aktivační funkce zahrnují:

- **ReLU (Rectified Linear Unit):** Rychlá a efektivní, často používaná ve skrytých vrstvách.
- **Sigmoid:** Tradičně používaná v dekódovací části pro hodnoty, které mají být mezi 0 a 1.
- **Tanh:** Používá se tam, kde jsou vstupní data normalizována mezi -1 a 1.

4.6.2 Trénink Autoencoderu

Autoencoder se trénuje pomocí algoritmu zpětného šíření chyby (backpropagation) ve spojení s optimalizačním algoritmem, jako je gradientní sestup. Proces zahrnuje několik kroků:

- **Forward Pass:** Výpočet výstupů sítě (rekonstrukce) a chyby vzhledem k očekávanému výstupu (vstupní data).

- **Backward Pass:** Výpočet gradientů funkce chyby vzhledem k vahám sítě, začínající od výstupní vrstvy a postupující zpět k vstupní vrstvě.
- **Aktualizace vah:** Váhy jsou aktualizovány tak, aby se minimalizovala rekonstrukční chyba sítě.

4.6.3 Použití Autoencoderu

Autoencodery mají široké spektrum použití včetně:

- **Redukce rozměrů:** Učení komprimovaných reprezentací vstupních dat pro snížení rozměrů.
- **Odstranění šumu:** Učení modelu, který dokáže odstranit šum z dat.
- **Generování dat:** Typ autoencoderu, jako je Variational Autoencoder (VAE), se používá k generování nových datových bodů.
- **Pretraining:** Autoencodery mohou být použity k předtrénování vrstev pro hluboké neuronové sítě.

Autoencodery jsou výkonné nástroje pro učení reprezentací, ale jejich efektivita závisí na správné architektuře včetně počtu vrstev, počtu neuronů ve vrstvách a volbě aktivačních funkcí. Obecně platí, že autoencodery mohou být velmi účinné pro mnoho úloh, ale mohou trpět některými omezeními, jako je přeučení a výpočetní náročnost, zejména při práci s velmi velkými datovými soubory nebo vysokou dimenzionalitou vstupů.

4.7 PCA

Analýza hlavních komponent (PCA z anglického *Principal Component Analysis*) je statistická technika používaná pro redukci dimenzí datové sady, při zachování co největšího množství informací. Cílem PCA je identifikovat směry, ve kterých data vykazují největší variabilitu, a tím zjednodušit komplexitu datového modelu bez zásadní ztráty informací.

PCA začíná výpočtem kovarianční matice datové sady, která poskytuje měřítko toho, jak jednotlivé proměnné závisí mezi sebou. Vlastní vektory (eigenvectors) této matice reprezentují hlavní komponenty dat, zatímco vlastní hodnoty (eigenvalues) určují míru variability, kterou každá komponenta popisuje [20].

Postup:

- **Standardizace dat**: Vstupní data jsou převedena na jednotkovou škálu (průměr 0 a směrodatná odchylka 1) pro každý atribut samostatně.
- **Výpočet kovarianční matice**: Kovarianční matice ukazuje, jak jsou jednotlivé proměnné vzájemně závislé.

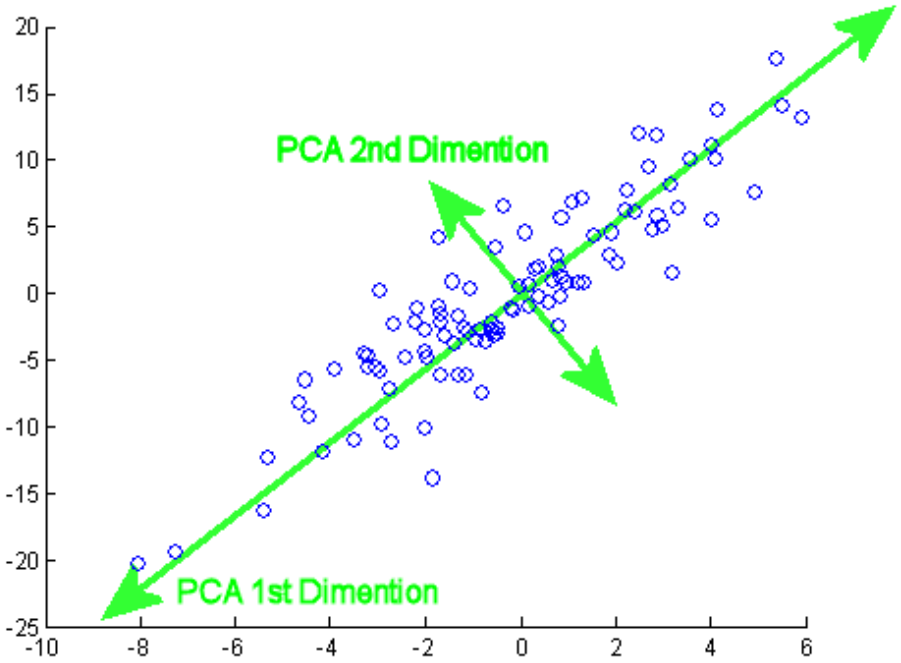
- **Výpočet vlastních vektorů a hodnot**: Určují hlavní osy dat a míru jejich variability.
- **Výběr komponent**: Počet hlavních komponent určíme podle kumulativního podílu vlastních hodnot, který pokrývá požadované procento celkové variability.

PCA se nejčastěji používá v těchto oblastech:

- Zjednodušení dat
- Odstranění šumu
- Optimalizace výpočetní náročnosti
- Zlepšení výkonu prediktivních modelů

Ačkoliv je PCA silný nástroj pro redukci dimenze, má také své nevýhody. Například, PCA je citlivá na měřítko proměnných, a proto je nutná předchozí standardizace. Navíc PCA předpokládá lineární vztahy mezi proměnnými a nemusí být vhodná pro data, která vykazují složité nebo nelineární vzory.

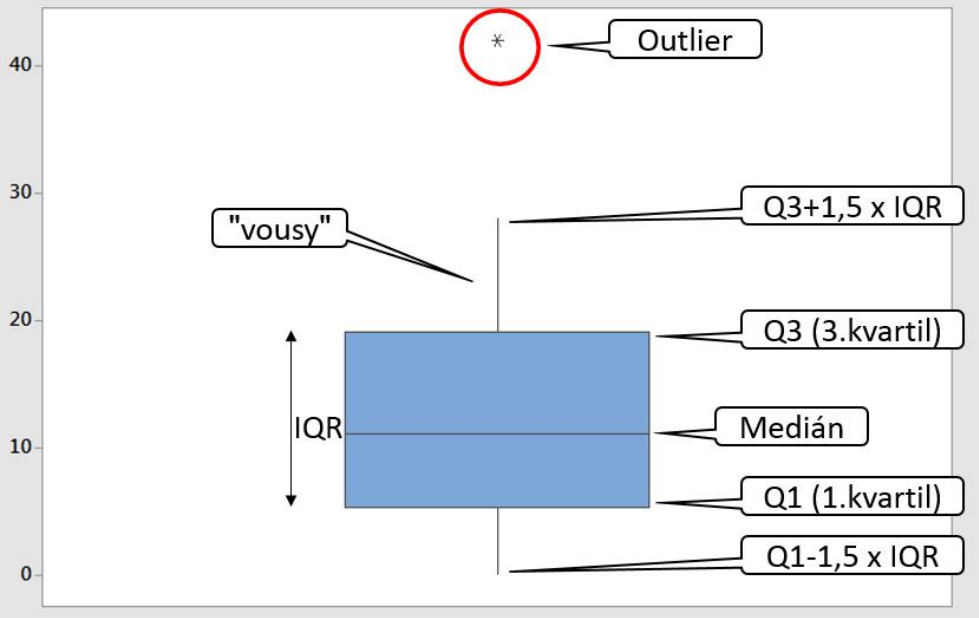
Analýza hlavních komponent je klíčovým nástrojem pro statistickou analýzu dat, který poskytuje užitečné vhledy při zpracování velkých datových sad. Díky své schopnosti redukovat dimenze dat při zachování podstatné části informací je neocenitelným nástrojem v mnoha oblastech.



Obrázek 3: Vizualizace hlavních komponent pomocí PCA [20]

4.8 Boxplot

Boxplot, známý také jako krabicový graf, je grafickým nástrojem používaným ve statistice pro znázornění distribuce datové sady. Skládá se z obdélníkového boxu, který zobrazuje mezikvartilový rozsah (rozložení středních 50 % dat) a z 'vousů' či čar, které ukazují rozptyl dat mimo kvartily. Linie uvnitř boxu označuje medián dat. Body ležící mimo vousy se často považují za odlehlé hodnoty a jsou zobrazeny samostatně. Tento graf poskytuje rychlý přehled o symetrii, rozložení a potenciálních odlehlých hodnotách v datové sadě [21].



Obrázek 4: Popis boxplotu [21]

5 Praktická část

Praktická část této práce byla realizována v programovacím jazyce Python, což umožnilo efektivní manipulaci a analýzu dat. Dataset, který byl použit, poskytla firma PTSW a obsahuje data z procesu válcování hliníku za studena. Po analýze dat došlo k hledání optimální kombinace předzpracování a regresního modelu vzhledem k MAPE a MSE.

5.1 Popis dat

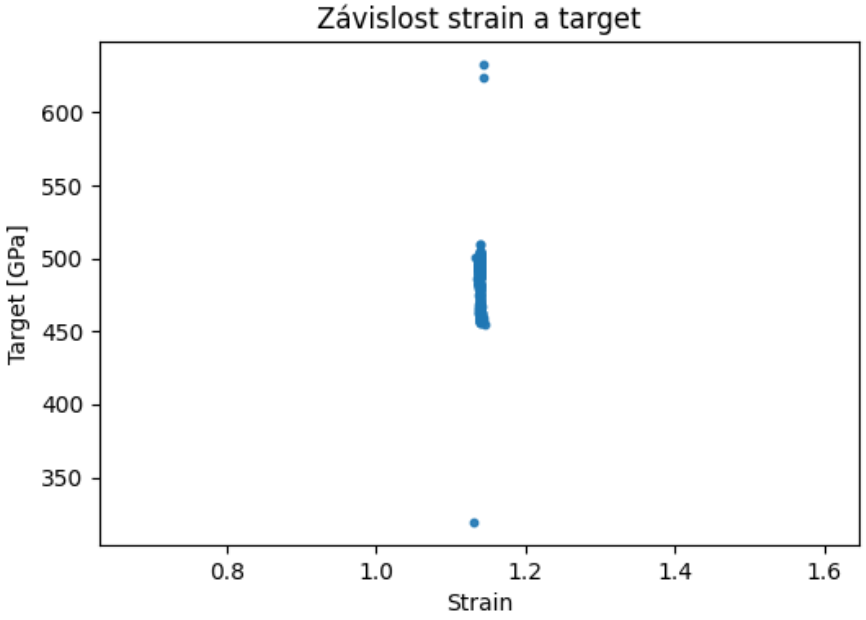
Společnost PTSW poskytla dataset zahrnující přes 500 000 vzorků dat, získaných během dvouměsíčního provozu. Data jsou uložena ve formátu XLSX. Každý záznam obsahuje následující atributy:

- PASS_ID: Jedinečný identifikátor průchodu.
- SEQ_NO: Sekvenční číslo v procesu.
- HEATING_SEQ: Číslo ohřevu v procesu.
- ALLOY: Typ slitiny.
- TYPE: Typ procesu.
- NN_TEMP: Teplota ve stupních Celsia (v tomto případě konstantní).
- NN_STRAIN: Logaritmus poměru vstupní a výstupní tloušťky.
- YIELD_STRENGTH_TARGET: Cílová hodnota tlaku vyvíjeného válcovacími hlavicemi.
- CHEMISTRY_INDX0 - CHEMISTRY_INDX9: Chemické indexy charakterizující procentuální složení materiálu (dohromady 10).
- CREATED: Datum a čas záznamu.

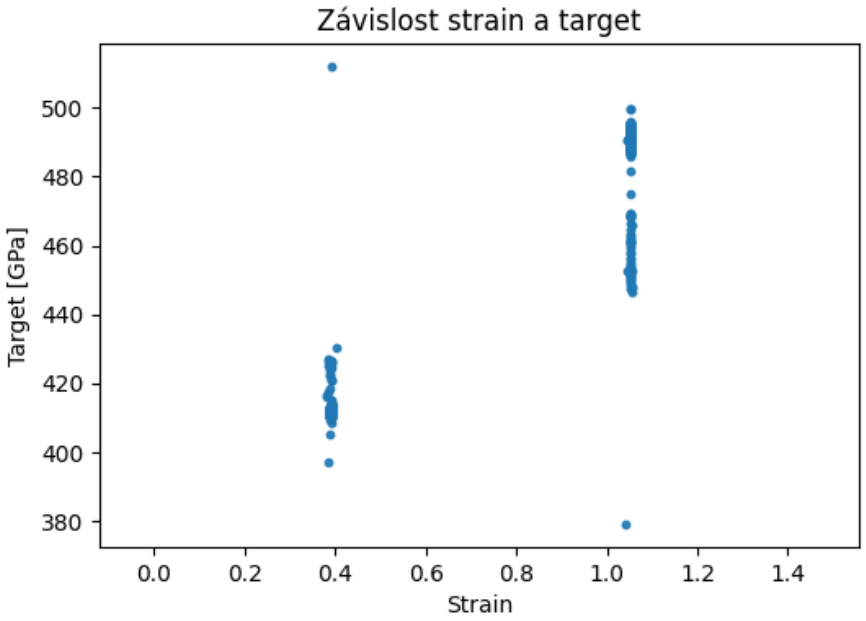
Cílem je predikce hodnoty YIELD_STRENGTH_TARGET (dále pouze *target*), jedná se tedy o regresní úlohu. Vstupní parametry pro modelování zahrnuje 11 proměnných: CHEMISTRY_INDX0 - CHEMISTRY_INDX9 a NN_STRAIN (dále pouze *strain*). Hlavní prvek ve slitinách je hliník (přes 94 %), dále je zastoupen například mangan, hořčík, křemík a měď.

Z důvodů časové efektivity a zjednodušení analýzy byl původní dataset omezen na 50 000 vzorků. Tento zredukovaný dataset obsahuje 72 různých kombinací chemického složení (slitin), přičemž téměř všechny kombinace mají jednu nebo dvě různé hodnoty proměnné *strain* a pouze dvě kombinace mají tři a čtyři různé hodnoty. Za stejnou hodnotu *strain* se považuje rozpětí $\pm$ 0.1.

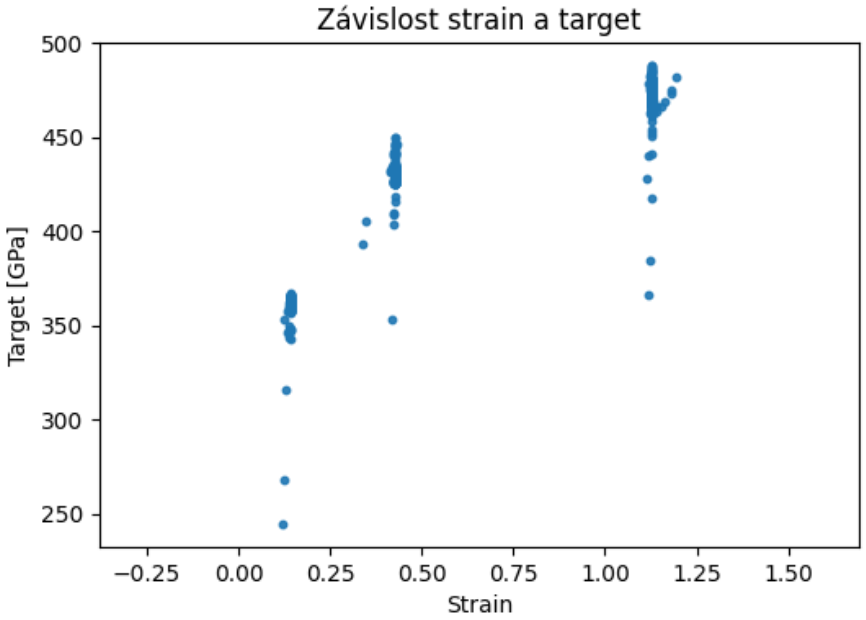
Analýza jednotlivých kombinací chemického složení je vizuálně prezentována na obrázcích, které ukazují všechna dostupná data (hodnoty *strain* a *target*) pro každou unikátní kombinaci. Každý obrázek poskytuje názorný přehled o rozložení hodnot v rámci dané kombinace, což pomáhá lépe porozumět variabilitě v datech a potenciálním vzorcům.



Obrázek 5: Příklad jedné známé hodnoty *strain* ve slitině

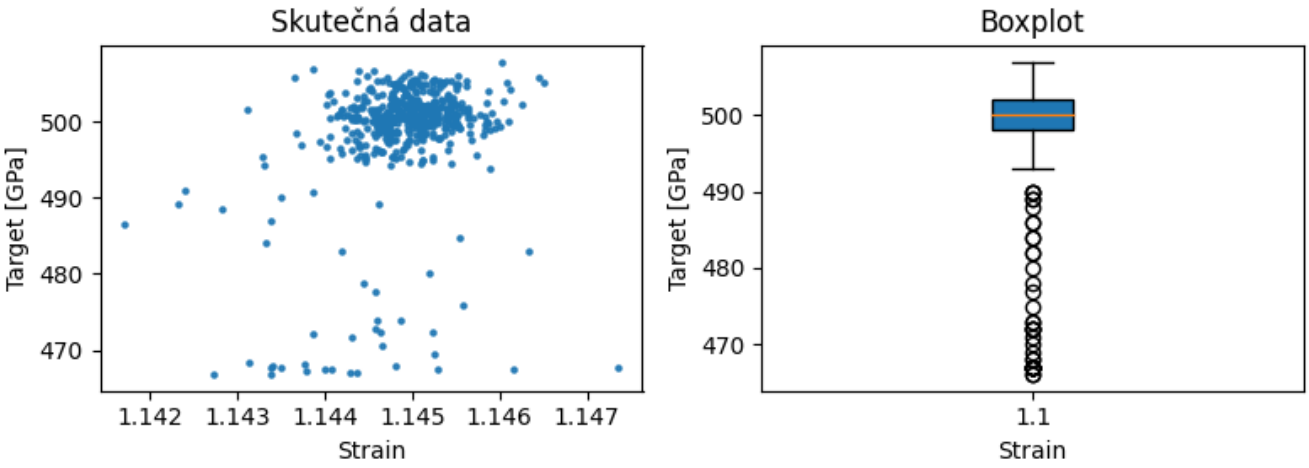


Obrázek 6: Příklad dvou známých hodnot *strain* ve slitině

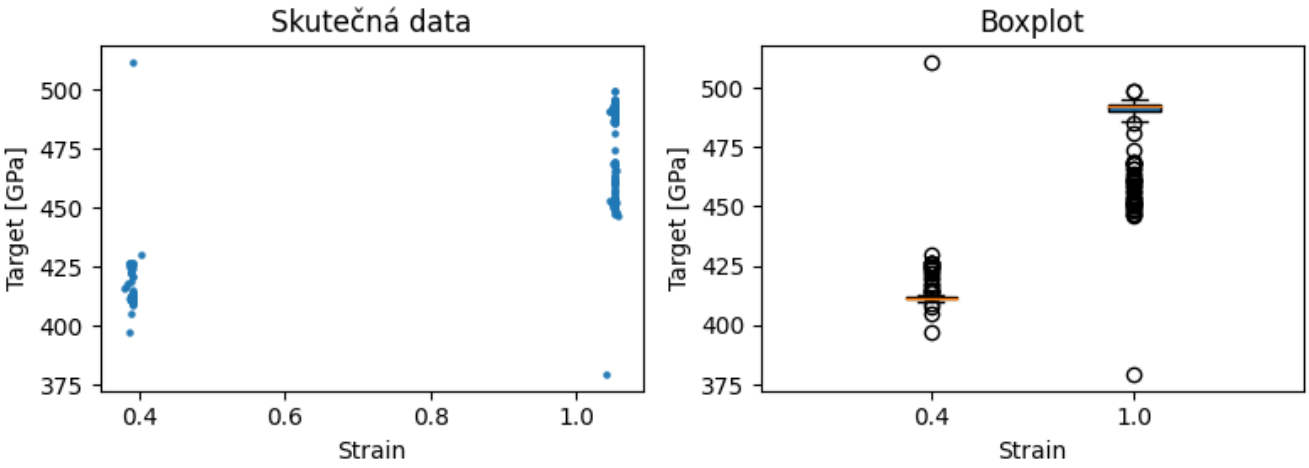


Obrázek 7: Příklad třech známých hodnot *strain* ve slitině

Kvůli nepřesnostem během sběru dat jsou *target* hodnoty zašumělé. Většinou jsou ale rozloženy blízko u sebe. To je znázorněno na následujících boxplotech.

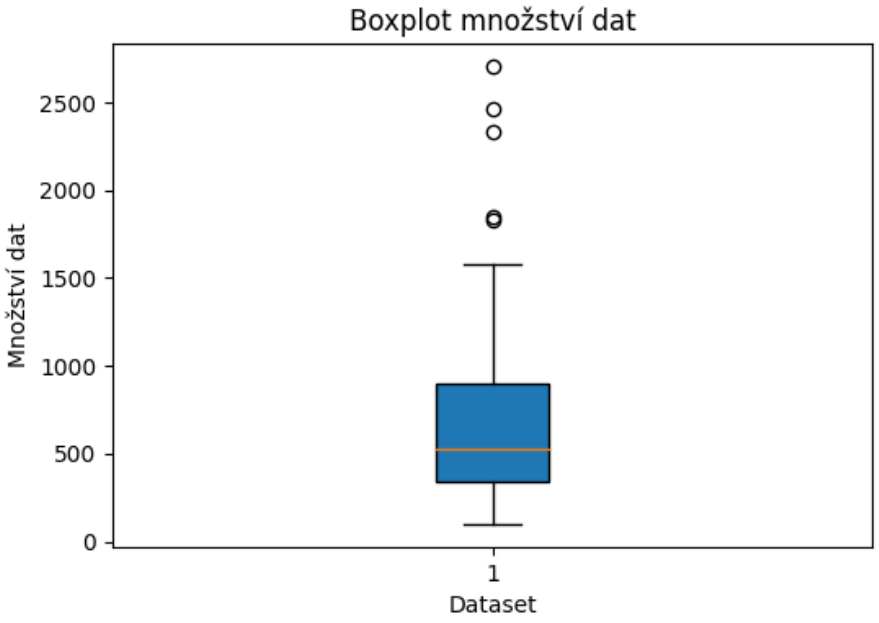


Obrázek 8: Příklad slitiny obsahující jednu hodnotu *strain*



Obrázek 9: Příklad slitiny obsahující dvě různé hodnoty *strain*

Každé unikátní chemické složení obsahuje v průměru 693 vzorků dat. Medián, reprezentovaný oranžovou čarou, činí 522. Většina dat se pak nachází v rozmezí od 344 do 899, což odpovídá hodnotám mezi prvním a třetím kvartilem a jsou zobrazeny v modrém boxu. Graf dále obsahuje tzv. *vousy*, které se táhnou od horního a dolního okraje boxu k nejvzdálenějším bodům, které jsou ještě považovány za běžné. Mimo oblast vousů se nachází odlehlé hodnoty (outliers), které výrazně vybočují z běžného rozložení dat.

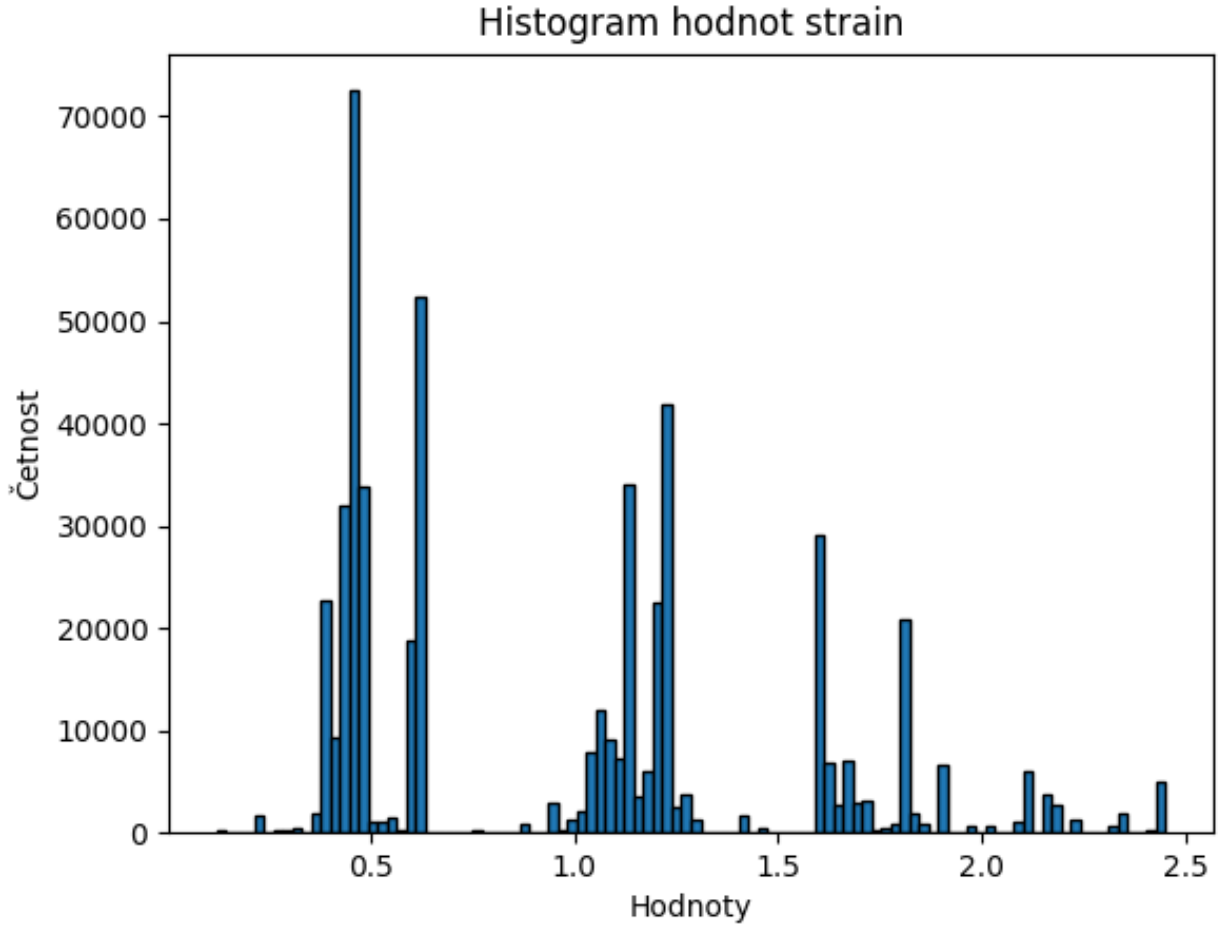


Obrázek 10: Množství dat obsahující jednotlivé jedinečné kombinace chemického složení

5.2 Definice úlohy

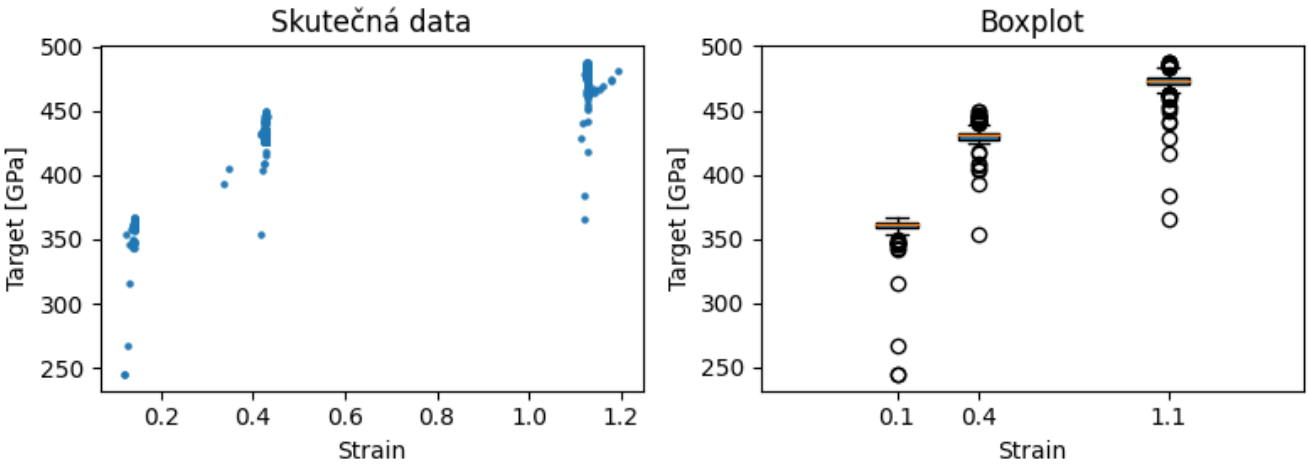
Cílem této práce je vyvinout co nejúčinnější model strojového učení pro předpověď hodnoty *target*. Tento model musí splňovat specifické požadavky válcovacího procesu, kde výsledná predikce nejen poskytuje přesné odhady, ale je také reprezentována funkcí vhodnou pro integraci do řídící jednotky procesu. Preferovanou formou této funkce (podle PTSW) je polynom, kvůli jeho schopnosti efektivně aproximovat různé závislosti, i když možnosti zahrnují také například exponenciální, logaritmické nebo lomené funkce.

Polynomická funkce bude vytvořena na základě chemického složení a dostupných datových bodů, a musí optimálně předpovídat hodnotu *target* v pracovním rozsahu *strain* od 0.1 do 2.5. Tento rozsah byl stanoven na základě analýzy histogramu všech hodnot *strain*.



Obrázek 11: Histogram hodnot *strain*

Pro ilustraci, v případě jedné slitiny s jedinečnou kombinací chemického složení očekáváme, že výsledný polynom bude efektivně procházet skrze datové body [*strain*,*target*]: [0.1, 360], [0.4, 430] a [1.1, 475], jak je uvedeno na obrázku 12.



Obrázek 12: Slitina obsahující tři různé hodnoty *strain*

Kromě toho se tato práce zaměří na predikci hodnoty *target* pro slitiny, u kterých nejsou k dispozici žádné historické údaje a jsou známa pouze chemická složení. Absence datových bodů znamená, že nemáme přímé referenční body pro vývoj a validaci modelů, což představuje významnou výzvu pro modelování.

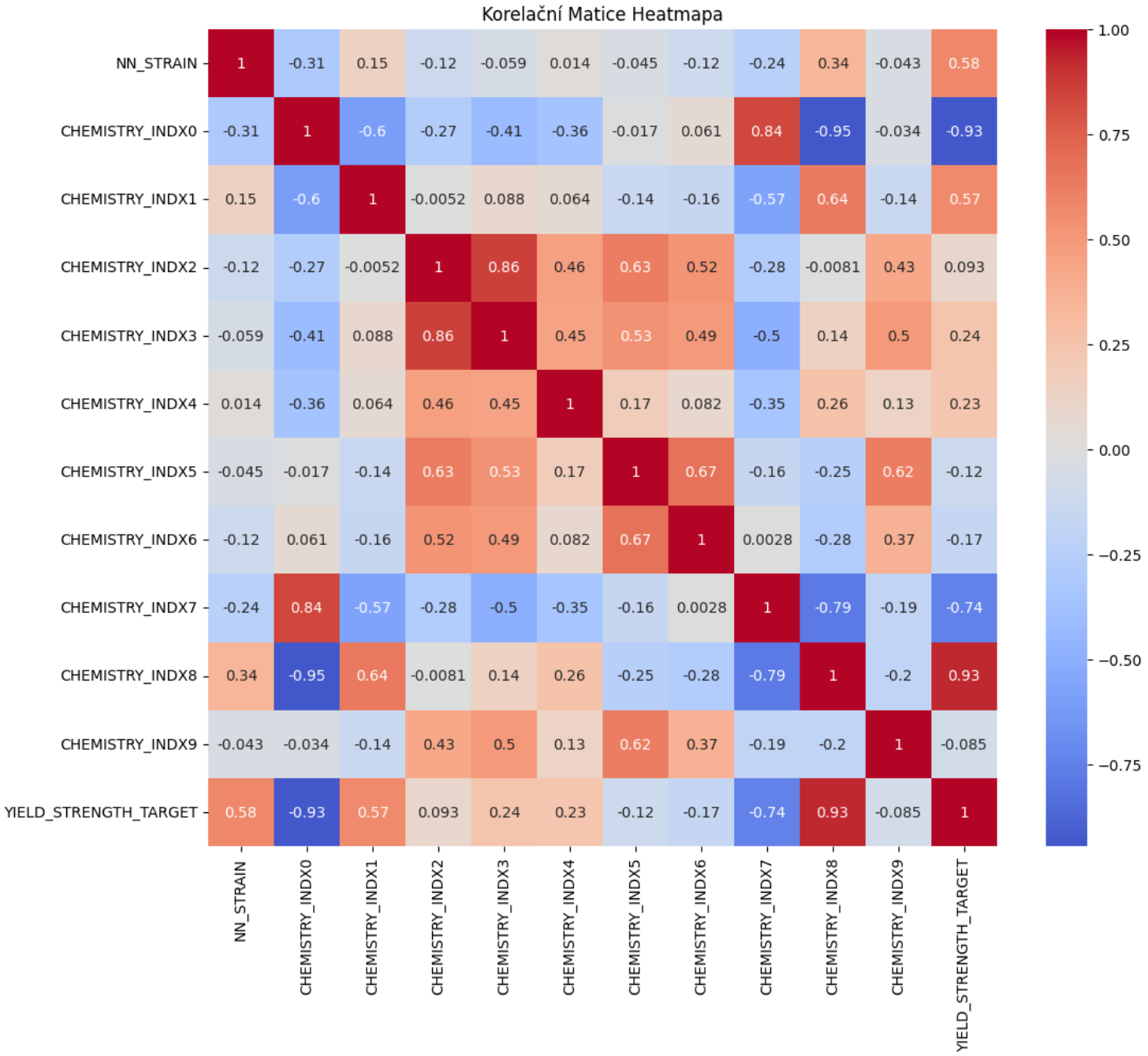
5.3 Datová analýza

Vzhledem k tomu, že data obsahují procentuální zastoupení různých chemických prvků, se dá předpokládat nelinearita dat.

V mnoha případech chemických složek, jako jsou slitiny, mohou existovat složité interakce mezi různými prvky. Tyto interakce často nejsou lineární. Například malá změna v procentuálním zastoupení jednoho prvku může mít výrazný vliv na výsledné vlastnosti materiálu, zatímco větší změny nemusí mít lineární působení.

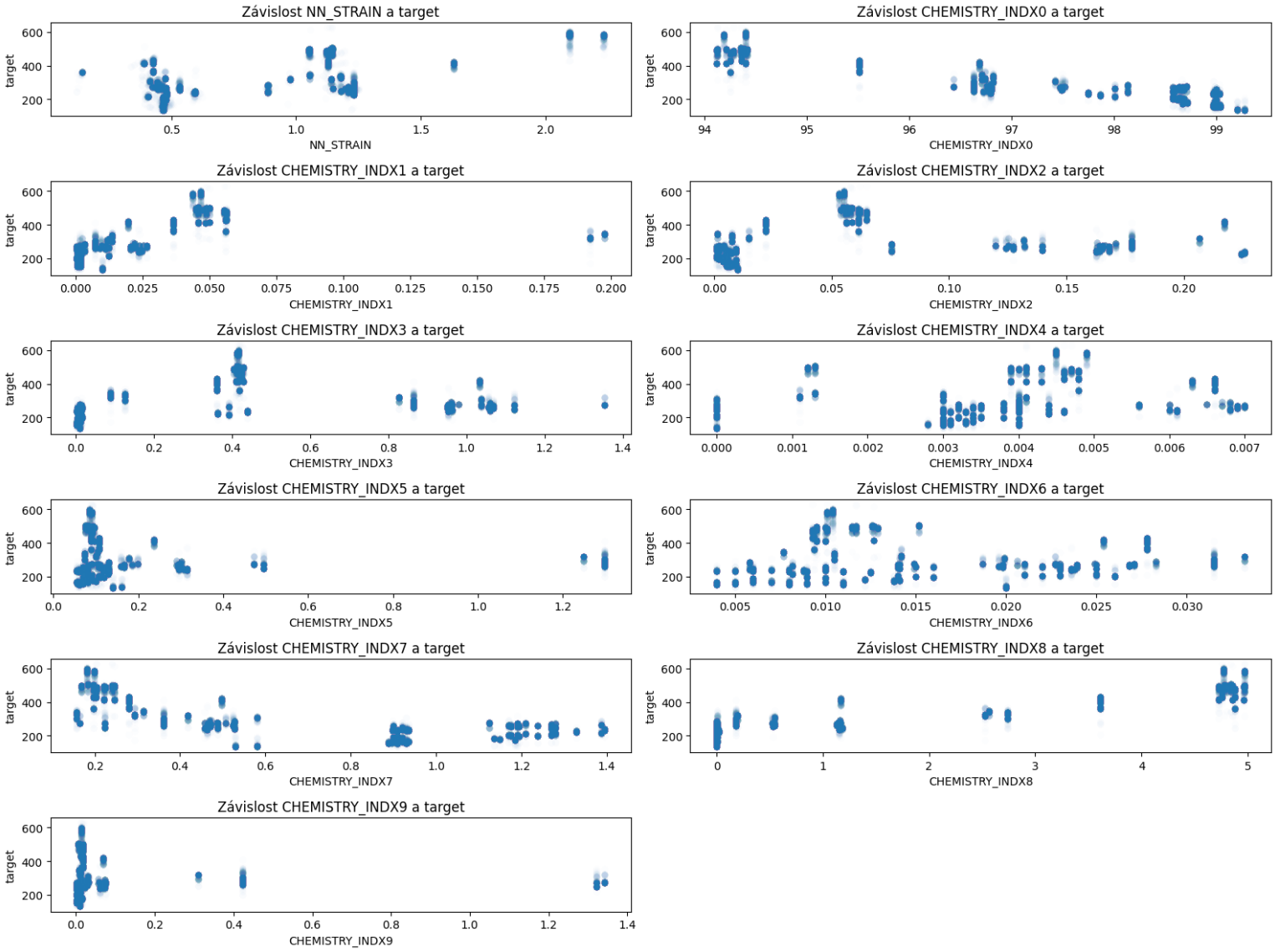
Procentuální hodnoty jsou omezeny na rozsah 0 – 100 %. To znamená, že proměnné jsou vzájemně závislé (součet všech procentuálních hodnot musí být 100 %), což může vést k nelineárním vzorcům.

Pro ověření nelinearity byla vytvořena *heatmap* [22] korelační matice příznaků a *target*.



Obrázek 13: Heatmap korelační matice

Zde je patrná silná lineární závislost příznaků $CHEMISTRY\_INDX0$ a $CHEMISTRY\_INDX8$ přesahující 90 %. Pro potvrzení jsou vykresleny všechny příznaky vzhledem k $target$.



Obrázek 14: Závislosti jednotlivých příznaků a *target*

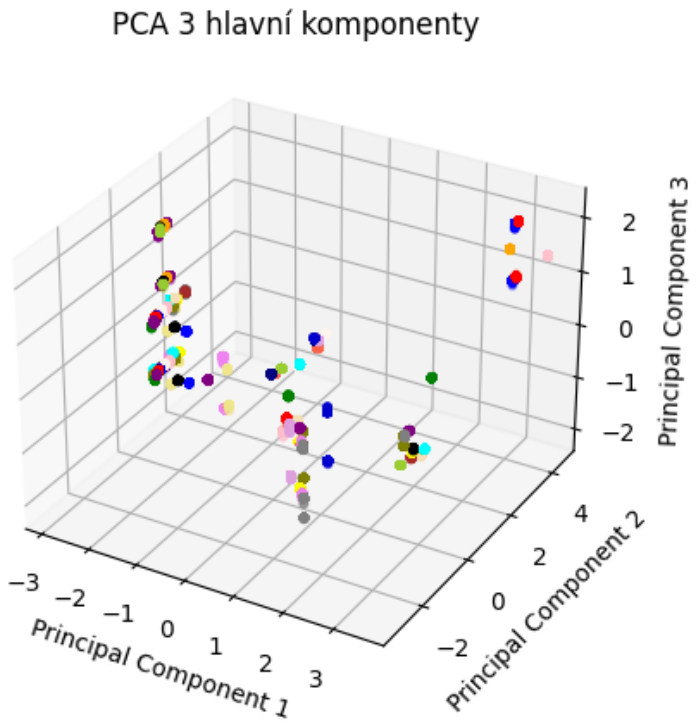
Je zřejmé, že výše zmíněné chemické indexy vykazují lineární závislost, zatímco ostatní indexy ukazují silnou nelinearitu. Celkově lze tedy data považovat za nelineární.

Protože neexistuje analytické řešení této úlohy a máme k dispozici dostatečné množství dat, je vhodné využít algoritmy strojového učení. Vzhledem k relativně nízké dimenzi příznaků (11) by měly stačit jednoduché modely.

5.3.1 PCA

Pro lepší pochopení vztahů a vzorů mezi daty byl vytvořen graf zobrazující PCA s třemi hlavními komponentami. V tomto 3D prostoru lze vizualizovat strukturu dat, což by v původním prostoru nebylo možné.

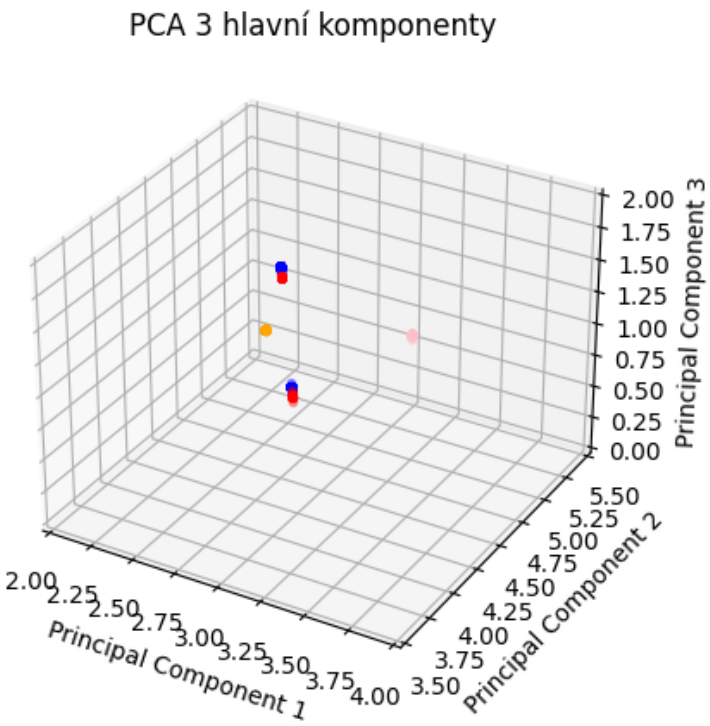
Každá jedinečná slitina byla obarvena jinou barvou. Všechny slitiny, které mají více různých hodnot *strain*, se tedy v tomto prostoru vykreslují na více stejně barevných bodech.



Obrázek 15: Vykreslení všech dat pomocí PCA

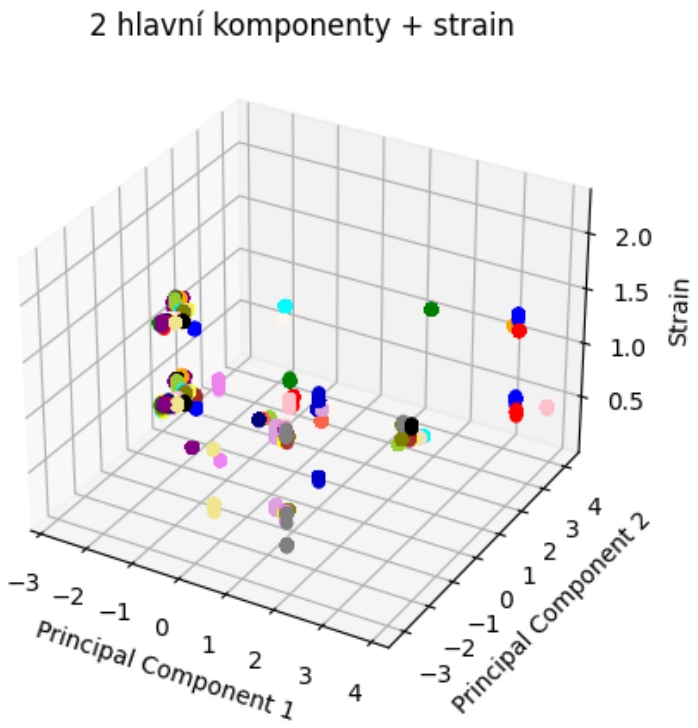
Z tohoto grafu lze vyvodit jaké body (slitiny) mají k sobě blízko a měli by tedy reagovat podobně. Také je zde vidět pár bodů vzdálených od ostatních. Je pravděpodobné, že pro tyto vzdálené body bude predikce méně přesná.

Pro lepší ilustraci je vykreslen pouze pravý horní roh, který tvoří oddělený shluk.



Obrázek 16: Vykreslení pouze části dat

Zde je vidět, že slitiny vykreslené modrou a červenou, mají podle chemického složení a hodnoty *strain* velice blízko k sobě. V jejich bezprostředním okolí se také nachází další dvě slitiny s pouze jednou známou hodnotou *strain*, tyto slitiny jsou reprezentované oranžovým a růžovým bodem.

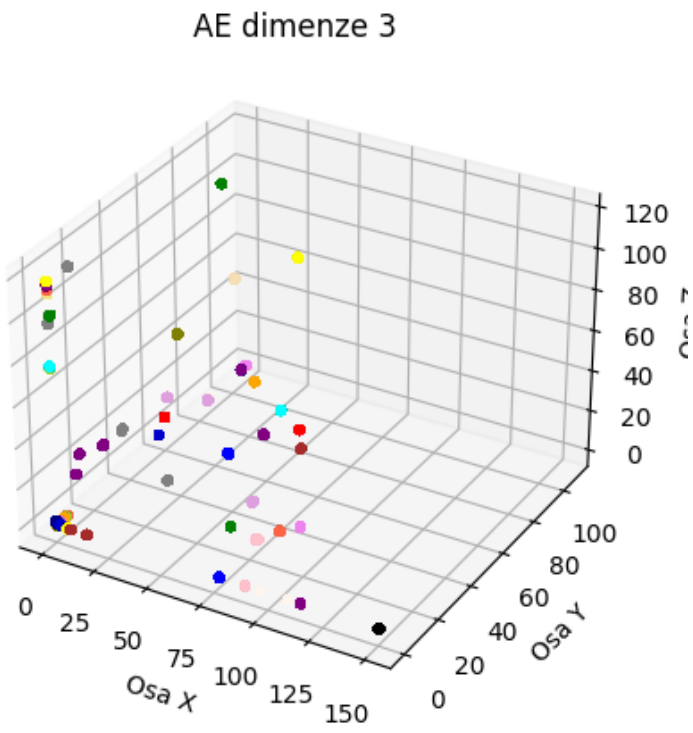


Obrázek 17: První a druhá hlavní komponenta s přidanou hodnotou *strain*

Další graf ukazuje data transformovaná pomocí PCA, kde byly vybrány pouze dvě hlavní komponenty, na vstupu bylo pouze chemické složení (bez *strain*). Třetí osa z znázorňuje hodnotu *strain* každé slitiny. Dá se konstatovat, že tento graf vykazuje velkou podobnost s grafem na obrázku 15. To naznačuje, že variabilita proměnné *strain* je podobná variabilitě třetí hlavní komponenty.

5.3.2 Autoencoder

Se stejným cílem, tedy pro lepší pochopení vzorů a vztahů v datech, byl natrénován autoencoder. Po jeho natrénování byla využita pouze jeho první část, tedy encoder, který převedl data do tří dimenzionálního latentního prostoru.



Obrázek 18: Encoded data do dimenze 3

Avšak výsledná reprezentace dat se zdá být nedostatečná. Předpokládalo se, že slitiny s různými hodnotami parametru *strain* by se ve vizualizaci měly shlukovat blízko sebe. Nicméně, podle získaných výsledků se zdá, že k očekávanému shlukování nedochází. Toto zjištění naznačuje potřebu dalšího zkoumání a možného přepracování modelu.

5.4 Způsob vyhodnocování

Při analýze optimální kombinace předzpracování a regresních modelů se klade zásadní důraz na volbu metody hodnocení prediktivní schopnosti modelu. Běžně aplikovaná metodika - **holdout** validace [23], zahrnuje rozdělení datové sady na 80 % trénovacích a 20 % testovacích dat. Tento přístup však neposkytuje adekvátní řešení pro situace, kdy nedochází k dostatečnému pokrytí datových bodů pro nové kategorie, jako jsou například nové slitiny bez historických dat o jejich chování.

V těchto případech se doporučuje uplatnit alternativní metodu, a sice metodu cross-validace typu **k-fold** [23]. Tato technika předpokládá náhodné rozdělení datové sady na K podskupin, přičemž model je iterativně trénován na K-1 podskupinách, zatímco zbývající podskupina slouží jako testovací vzorek. Tento proces se opakuje K-krát, takže každá podskupina je přesně jednou využita jako testovací. Výsledné metriky jsou následně zprůměrovány pro získání robustního odhadu výkonnosti modelu.

V této úloze je ale vhodné použít modifikovanou verzi této metody, kde dataset není rozdělen na K podmnožin, ale na počet unikátních chemických kombinací přítomných v da-

tové sadě. Každá takováto jedinečná podmnožina pak reprezentuje všechna data příslušející specifické chemické kombinaci. Tento přístup umožňuje realističtější simulaci situace, ve které se do výrobního procesu začleňuje úplně nová slitina. Díky tomu je možné lépe odhadnout, jak model zareaguje na opravdu neznámá data. Hodnocení výkonnosti modelu bude prováděno stejnými metrikami, jaké jsou uvedeny v kapitole 5.5, konkrétně se jedná o střední kvadratickou chybu (MSE) a průměrnou procentuální absolutní chybu (MAPE). Přičemž MAPE bude přikládána větší váha než MSE.

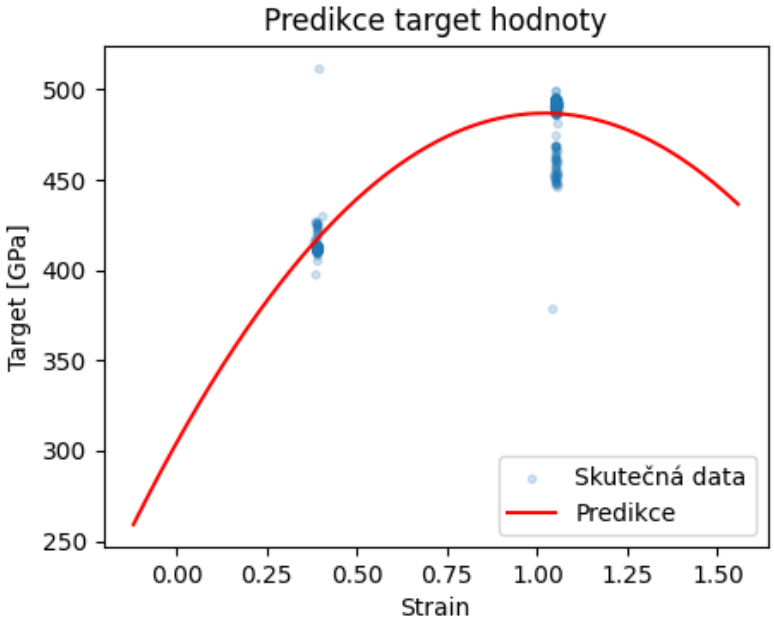
Pro zajištění co nejrelevantnějších výsledků budou modely validovány oběma metodami: **holdout** a **k-fold**, aby bylo možné porovnat jejich efektivitu a přesnost v různých scénářích.

5.5 Přístup PTSW

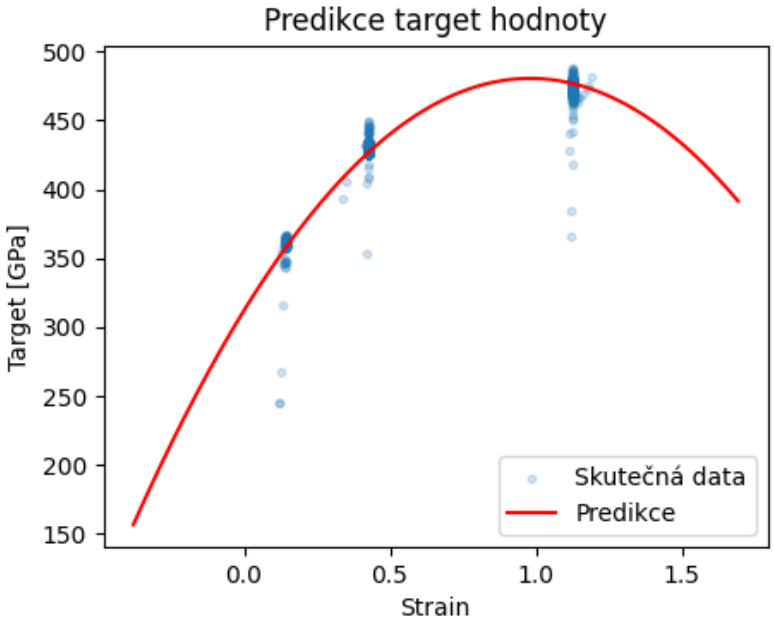
Firma PTSW zpřístupnila jeden ze svých rozpracovaných modelů. Data jsou rozdělena na 80 % trénovací a 20 % testovací sady (holdout). Model je strukturován jako pipeline, kde je nejprve aplikováno MinMax škálování a následně polynomiální předzpracování se stupněm 2. Na závěr je model trénován pomocí algoritmu lineární regrese. S tímto modelem bylo dosaženo průměrné procentuální odchylky (MAPE) **2.23 %** a střední kvadratické chyby (MSE) **110.5** napříč všemi daty. Jedná se tedy o velice dobré výsledky.

I když jsou vztahy mezi vstupními a výstupními proměnnými nelineární, použití tohoto přístupu umožňuje zachytit složité vzorce v datech. Obvykle lineární regrese vyžaduje lineární vztahy mezi proměnnými pro optimální funkčnost. Avšak aplikace nelineární transformace dat, konkrétně polynomiálního předzpracování, tento problém efektivně řeší.

Následující grafy zobrazují predikce cílové hodnoty *target* tímto modelem. Pro zlepšení přehlednosti byly limity predikcí upraveny tak, aby se pohybovaly od minimální hodnoty *strain* snížené o 0.5 až po maximální hodnotu *strain* zvýšenou o 0.5. Aby bylo možné lépe rozlišit rozložení skutečných dat, byla nastavena jejich průsvitnost na 20 %. Toto nastavení umožňuje jasněji identifikovat oblasti s vyšší hustotou dat a tedy zobrazit, kudy by měla prediktivní křivka ideálně procházet.



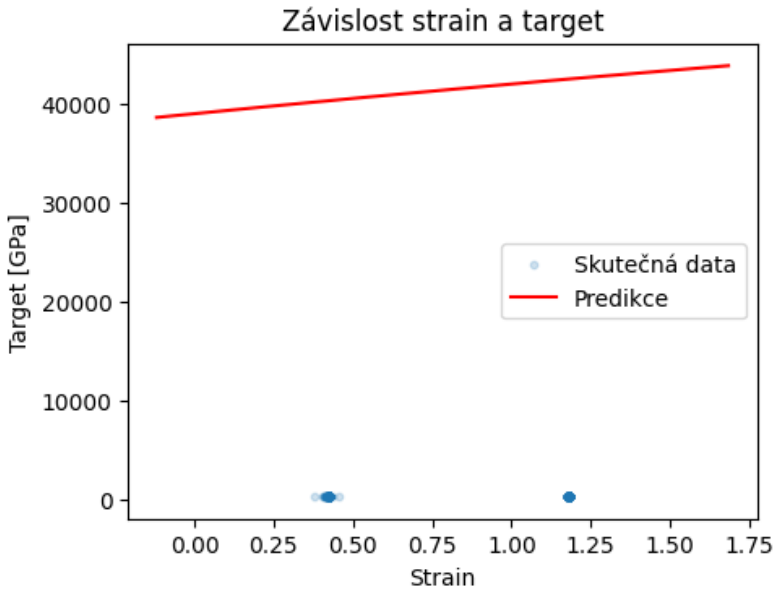
Obrázek 19: Predikce *target* slitiny č. jedna s rozšířeným rozsahem modelem PTSW



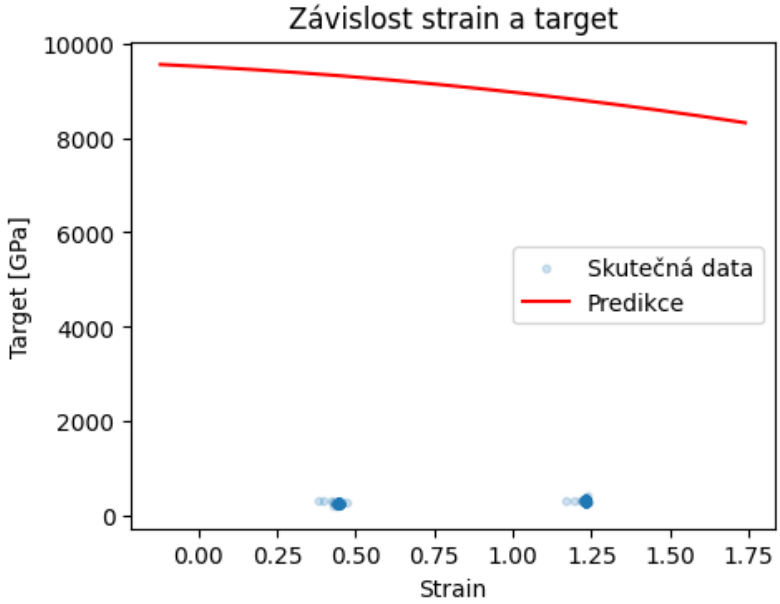
Obrázek 20: Predikce *target* slitiny č. pět s rozšířeným rozsahem modelem PTSW

Na obrázku 19 dosáhla predikce MSE: 96.9 a MAPE: 1.50 %. Na obrázku 20 MSE: 52.8 a MAPE: 0.99 %.

Při použití k-fold validace, ale tento model od PTSW těžce selhává a dosahuje v průměru přes všechny slitiny hodnot MAPE **428.5 %** a MSE **34288949.2**. Pravděpodobně došlo vzhledem k vysoké složitosti modelu k přeučení. Model vykazuje extrémní predikce, jako třeba:



Obrázek 21: Predikce *target* slitiny č. 17 s rozšířeným rozsahem



Obrázek 22: Predikce *target* slitiny č. 15 s rozšířeným rozsahem

Tyto konkrétní slitiny zaznamenaly MAPE: 8843.08 % a 15587.34 %. Celkově je tedy vhodné hledat jiný model, protože i když při holdout validaci byla odchylka minimální, tak při simulaci zavedení nových slitin model selhává.

6 Předzpracování a regresní modely

Vzhledem k vysokým výpočetním nárokům byla následující analýza provedena na zmenšeném datasetu. Byly testovány různé kombinace technik předzpracování dat a regresních modelů. Hlavní metrika pro hodnocení výkonu byla firmou PTSW zvolena MAPE. Pro další analýzu strojového učení jsme se rozhodli zaměřit na následující **regresní modely**:

- KNeighborsRegressor (kNN),
- GradientBoostingRegressor,
- XGBRegressor,
- RandomForestRegressor,
- DecisionTreeRegressor,
- ExtraTreesRegressor,
- MLPRegressor,
- LinearRegression.

U MLP modelu byla nastavena jedna skrytá vrstva s 317 neurony a obsahuje tedy 4122 trénovatelných parametrů. U všech ostatních byly nastaveny defaultní (výchozí) parametry.

kNN:

- $n\_neighbors = 5$ - počet sousedů, které algoritmus bere v úvahu při výpočtu,
- $weights =$ 'uniform' - stejná váha všem sousedům.

GradientBoostingRegressor:

- $loss =$ 'squared_error' - funkce ztráty,
- $learning\_rate = 0.1$,
- $n\_estimators = 100$ - počet boostingových fází.

XGBRegressor:

- *objective* = 'reg:squarederror' - funkce ztráty,
- *base_score* = 0.5 - počáteční predikce.

RandomForestRegressor:

- *criterion* = 'squared_error' - funkce ztráty,
- *n_estimators* = 100 - počet stromů v lese.

DecisionTreeRegressor:

- *criterion* = 'squared_error' - funkce ztráty,
- *max_depth* = None - maximální hloubka stromu.

ExtraTreesRegressor:

- *criterion* = 'squared_error' - funkce ztráty,
- *n_estimators* = 100 - počet stromů v lese.

LinearRegression:

- *fit_intercept* = True - vypočítání průsečíku (intercept).

Použité metody předzpracování dat:

- Inverzní transformace,
- Yeo-Johnson transformace se standardizací,
- Logaritmická transformace,
- Standardizace,
- Normalizace.

6.1 Holdout cross-validace

Nejprve budou tyto algoritmy testovány metodou holdout, při stejném rozdělení dat jako v poskytnutém modelu od PTSW, tedy 80 % trénovacích a 20 % testovacích. Nicméně toto rozdělení nebude zcela náhodné (tak jako tomu bylo u přístupu PTSW), ale dojde k rozdělení každé slitiny na 80 % trénovací a 20 % testovací. Tím je zajištěno, že každá slitina bude dostatečně zastoupena v trénovacích datech. Tabulka 1 zobrazuje výkony všech testovaných modelů při inverzní transformaci dat. Tabulka 2 zase ukazuje výkony při použití logaritmické transformace dat. Zároveň jsou zvýrazněny nejlepší dosažené hodnoty.

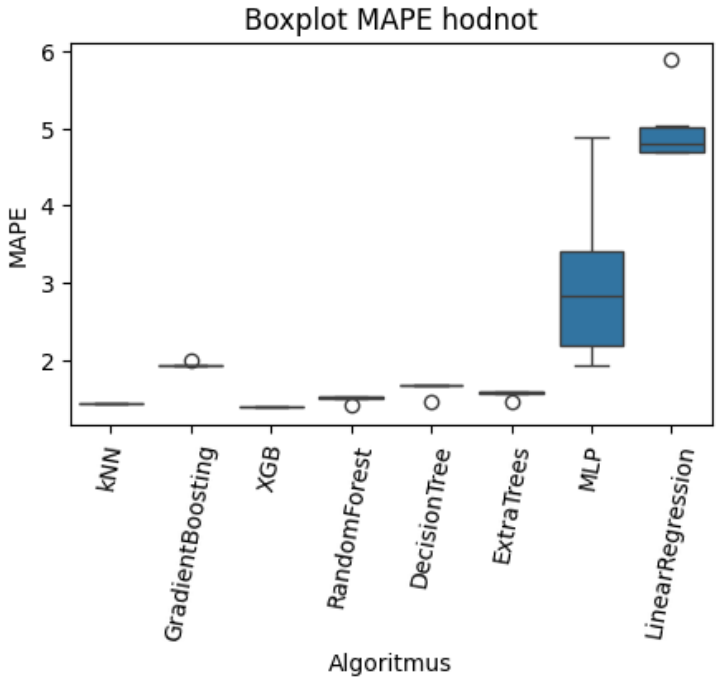
Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.28
GradientBoostingRegressor	1.94 %	88.97
XGBRegressor	**1.40 %**	**68.43**
RandomForestRegressor	1.52 %	83.23
DecisionTreeRegressor	1.69 %	108.75
ExtraTreesRegressor	1.59 %	95.16
MLPRegressor	3.51 %	189.95
LinearRegression	5.88 %	621.28

Tabulka 1: Srovnání výkonů modelů při inverzní transformaci

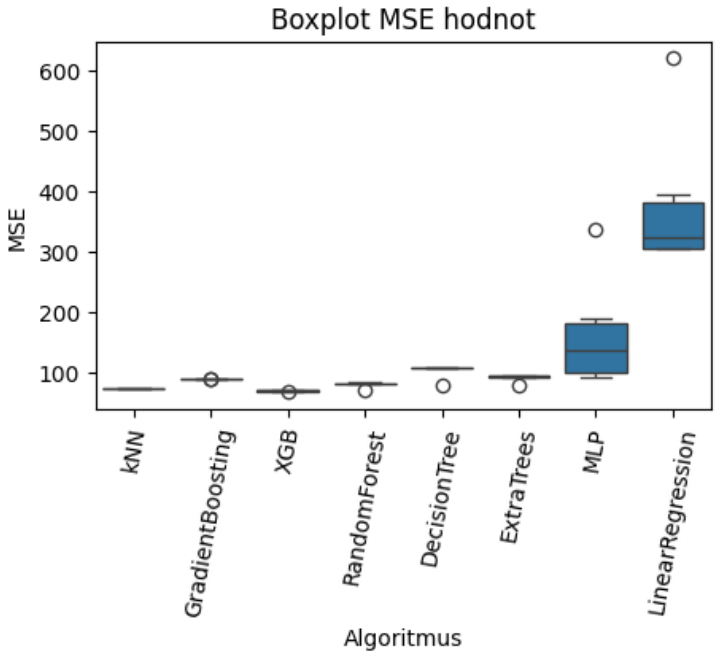
Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.30
GradientBoostingRegressor	1.94 %	89.09
XGBRegressor	**1.41 %**	**70.58**
RandomForestRegressor	1.53 %	83.96
DecisionTreeRegressor	1.69 %	107.61
ExtraTreesRegressor	1.59 %	94.20
MLPRegressor	2.55 %	119.98
LinearRegression	4.90 %	343.47

Tabulka 2: Srovnání výkonů modelů při logaritmické transformaci

Kompletní výsledky jsou podrobně dokumentovány v příloze 10. Pro zlepšení přehlednosti jsou výsledky zobrazeny pomocí boxplotů, které ilustrují výkony jednotlivých algoritmů při různých metodách předzpracování dat.



Obrázek 23: Boxploty výkonů modelů (MAPE) při různých předzpracování dat



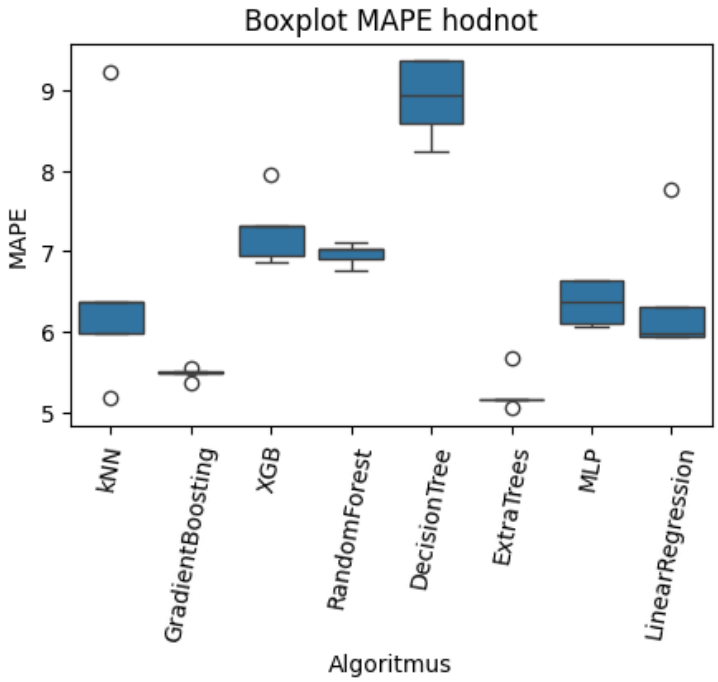
Obrázek 24: Boxploty výkonů modelů (MSE) při různých předzpracování dat

Z těchto boxplotů je zřejmé, že modely rozhodovacích stromů a kNN vykazují minimální variabilitu výkonu při různých metodách předzpracování a zároveň dosahují vynikajících výsledků. Nejlepší výkon předvedl model XGB s hodnotou MAPE 1.40 % a těsně za ním model kNN s hodnotou MAPE 1.45 %. Co se týče modelu MLP, nejúčinnějším předzpracováním se ukázala být Yeo-Johnson transformace, s níž dosáhl hodnoty MAPE 1.94 %. Model lineární regrese dosáhl hodnoty MAPE 4.69 % při standardizaci a také při normalizaci dat.

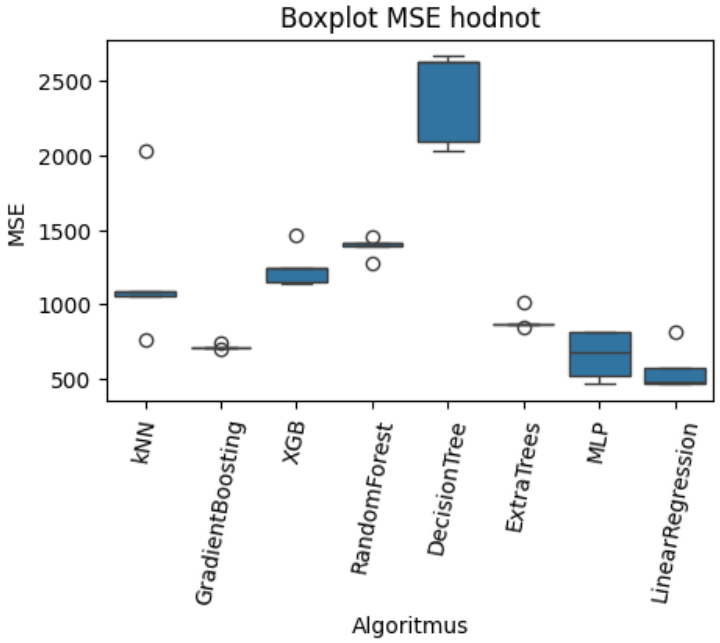
6.2 k-fold cross-validace

Data byla rozdělena podle metodiky uvedené v kapitole 5.4. Většina modelů vykazovala výrazně horší výsledky při použití normalizace dat (zobrazeno v tabulce 3), s MAPE hodnotami v rozmezí 11-73 %, proto byly tyto výsledky vynechány z obou boxplotů, aby byla zvýrazněna lepší výkonnost ostatních předzpracování. Výjimkou je model MLP, který s normalizací dosáhl nejen dobrých, ale dokonce svých nejlepších výsledků s MAPE na úrovni 5.92 %.

Je poměrně neobvyklé, aby normalizace měla negativní dopad na většinu testovaných algoritmů. Tento jev je pravděpodobně způsoben ztrátou informace. V případě našich dat jsou některé vstupní proměnné klíčové pro predikci nejspíše právě kvůli svému většímu rozsahu. Když jsou tyto proměnné normalizovány, jejich rozsah je zmenšen na úroveň méně důležitých proměnných. Tím se snižuje jejich vliv na model, což může vést ke ztrátě kritické informace a následně k horším výkonům modelů



Obrázek 25: Boxploty výkonů modelů (MAPE) při různých předzpracování dat



Obrázek 26: Boxploty výkonů modelů (MSE) při různých předzpracování dat

V analýze výsledků je patrný výrazný rozdíl v dosažených hodnotách. S ohledem na význam metriky MAPE se podíváme blíže na příslušný boxplot. Z něj vyplývá, že nejlepšího výsledku dosáhl model Extra Trees s použitím Yeo-Johnson transformace dat (tabulka 4), kde MAPE dosáhlo hodnoty 5.05 %. Tento model se navíc ukázal jako velmi robustní vůči různým metodám předzpracování dat. Velkou robustnost a dobré výsledky vykázal také model GradientBoosting. Zaslouží si pozornost i model kNN, který s logaritmickým předzpracováním dat dosáhl výsledku MAPE 5.17 %.

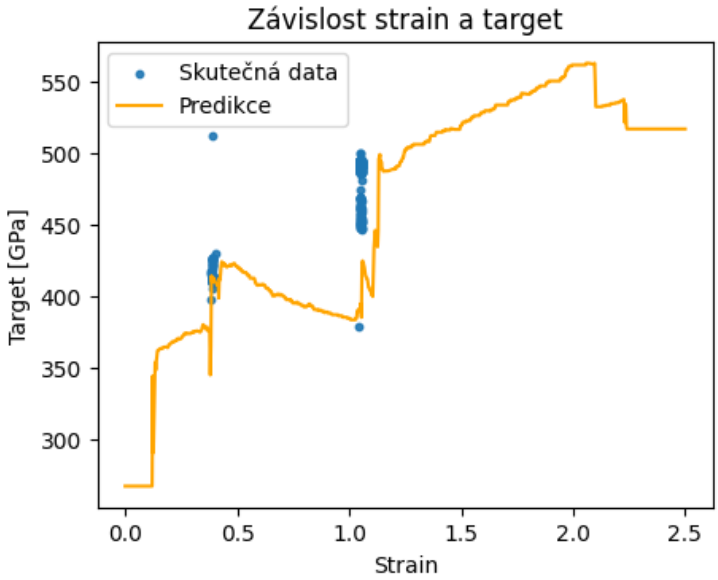
Metoda	MAPE	MSE
KNeighborsRegressor	20.65 %	8885.90
GradientBoostingRegressor	14.04 %	4160.86
XGBRegressor	20.18 %	14963.96
RandomForestRegressor	12.59 %	3484.49
DecisionTreeRegressor	12.18 %	3029.67
ExtraTreesRegressor	11.72 %	3777.98
MLPRegressor	**5.92 %**	**499.07**
LinearRegression	73.56 %	111218.41

Tabulka 3: Srovnání výkonů modelů při normalizaci dat

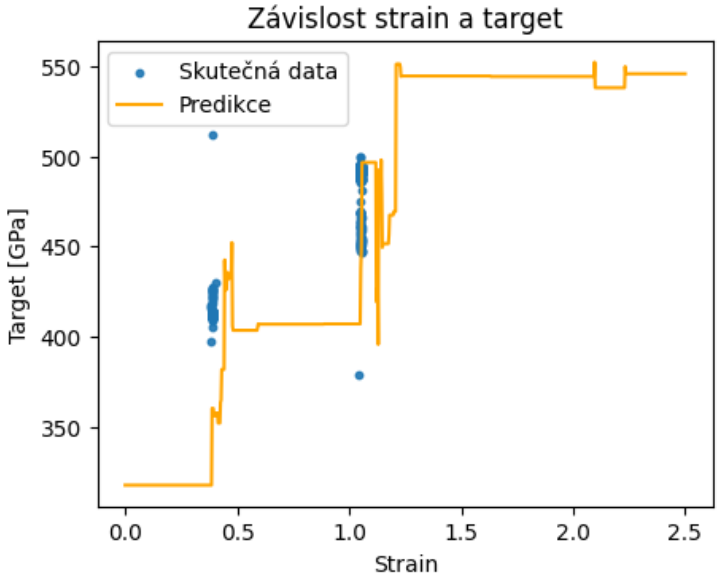
Metoda	MAPE	MSE
KNeighborsRegressor	5.98 %	1052.79
GradientBoostingRegressor	5.50 %	712.85
XGBRegressor	6.86 %	1137.35
RandomForestRegressor	6.76 %	1274.93
DecisionTreeRegressor	8.60 %	2033.31
ExtraTreesRegressor	**5.05 %**	844.11
MLPRegressor	6.06 %	**541.31**
LinearRegression	6.30 %	571.52

Tabulka 4: Srovnání výkonů modelů při Yeo-Johnson transformaci dat

Kompletní tabulky se pak nachází v příloze 11.

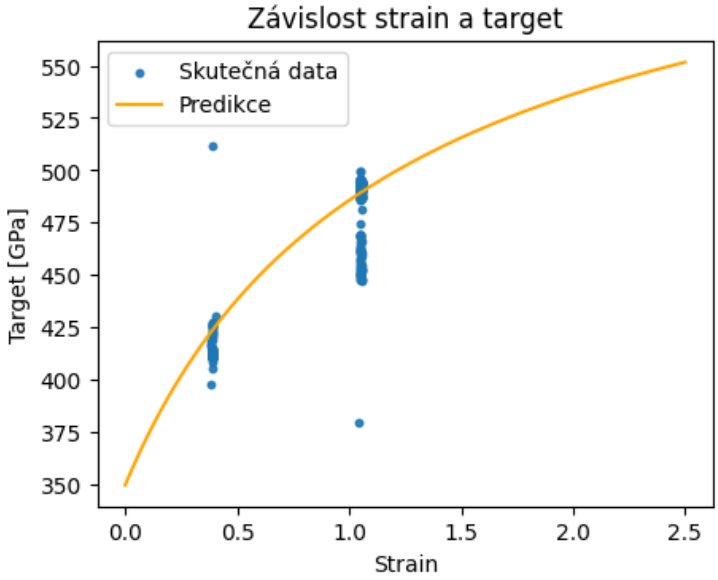


Obrázek 27: Příklad predikce slitiny č. jedna modelem Extra Trees

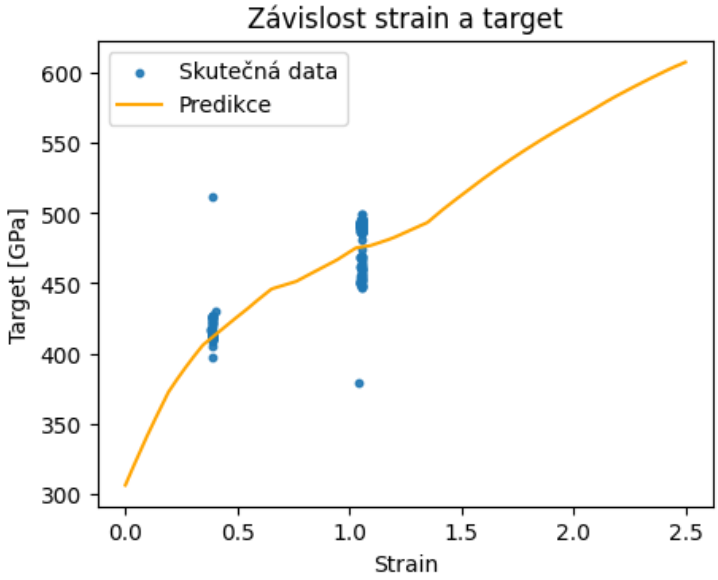


Obrázek 28: Příklad predikce slitiny č. jedna modelem XGB

Na obrázcích 27 a 28 jsou patrné ostré skoky a změny v predikovaných hodnotách, predikce tedy nejsou hladké. Tato charakteristika může komplikovat pozdější použití polynomiální aproximace.



Obrázek 29: Predikce *target* slitiny č. jedna modelem Lineární regrese



Obrázek 30: Predikce *target* slitiny č. jedna modelem MLP

Naopak, na obrázcích 29 a 30 lze vidět, že predikce jsou hladké, což značně usnadňuje možnost polynomiální aproximace. Přestože tyto modely obecně dosahují horších výsledků než ostatní, nabízí větší potenciál pro plynulou aproximaci.

7 Aproximace polynomem

Predikce získané z vybraných modelů budou nyní podrobeny aproximaci, aby se získal finální přehled o efektivitě každého modelu. Aproximace bude provedena pomocí polynomu třetího řádu s využitím metody lineární regrese. V této fázi se zaměříme pouze na modely: kNN, MLP, Extra Trees a lineární regrese.

U modelu kNN (k-nearest neighbors) došlo k ruční optimalizaci hyperparametrů. To znamená, že parametry modelu, jako je počet sousedů (*n_ neighbors*), způsob vážení sousedů (*weights*) a metrika vzdálenosti (*metric*), byly upraveny a testovány manuálně s cílem najít kombinaci, která poskytuje nejlepší výsledky predikce (MAPE) na testovacích datech. Tato optimalizace byla provedena bez automatických metod.

U modelu Extra Trees byla optimalizace hyperparametrů realizována pomocí bayesovské optimalizace (využívá pravděpodobnostní distribuční funkce k efektivnímu nalezení optimálních hyperparametrů modelu [11]), což vedlo ke značnému zrychlení výpočtů. Nicméně také přinesla mírné snížení výkonu a zvýšení variability výsledků. U modelu MLP došlo také k úpravě parametrů, zde ale pouze s cílem zrychlit výpočet. Takže byl především drasticky snížen počet iterací (epoch), čímž byl opět snížen výkon.

7.1 Provedení aproximace

Po získání rozšířené predikce (v rozsahu 0 až 2.5) dojde k polynomiální aproximaci této predikce třetím řádem. To znamená, že se pomocí lineární regrese naleznou optimální koeficienty polynomu:

$$y = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3, \tag{11}$$

kde a_0, a_1, a_2, a_3 jsou koeficienty naučené modelem a x zastupuje hodnotu *strain*. Následující kód ukazuje, zápis aproximace v jazyce Python.

```
X = np.linspace(0, 2.5, 1000)
y = y_pred

degree = 3

# Vytvoření a trénování polynomiálního regresního modelu
poly_model = make_pipeline(PolynomialFeatures(degree), LinearRegression())
poly_model.fit(X, y)

# Generování nových vstupních dat
X_fit = np.linspace(min(X), max(X), 1000).reshape(-1, 1)
# Predikce výstupních hodnot pro nová vstupní data
y_fit = poly_model.predict(X_fit)
```

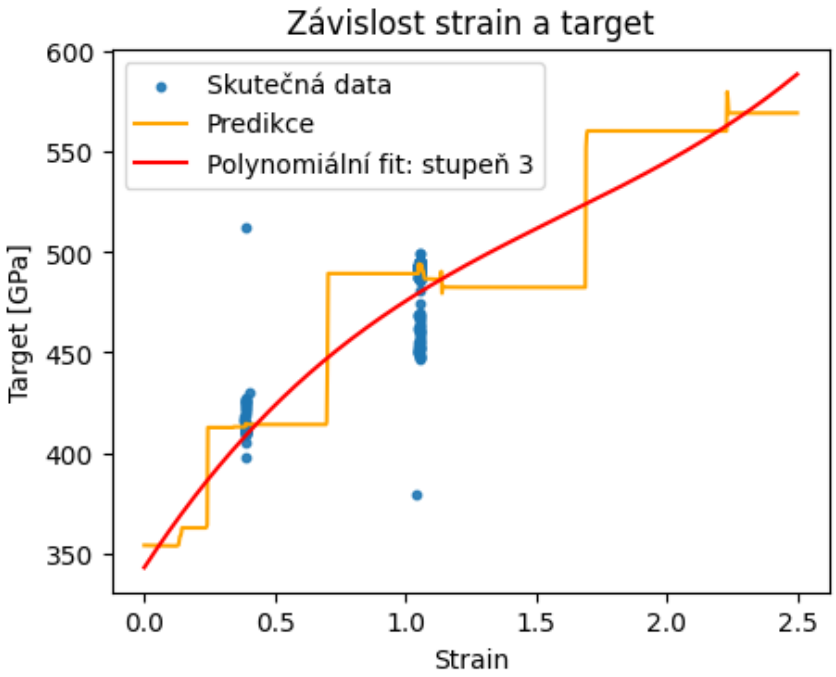
Výsledky nejslibnějších modelů jsou ukázány v tabulce 5. V té je vidět, že nejlepšího výkonu dosáhl model lineární regrese se standartizací.

Model	**MAPE**	**MAPE aproximace**
kNN s logaritmickou transformací	4.36 %	6.65 %
MLP s normalizací	6.40 %	6.45 %
Extra Trees s Yeo-Johnson transformací	5.28 %	7.31 %
Lineární regrese se standardizací	5.94 %	**5.94 %**

Tabulka 5: Srovnání výkonu modelů na zmenšeném datasetu

Následující obrázky ukážou, jak vypadá predikce a následná polynomiální aproximace modelů u konkrétní (první) slitiny.

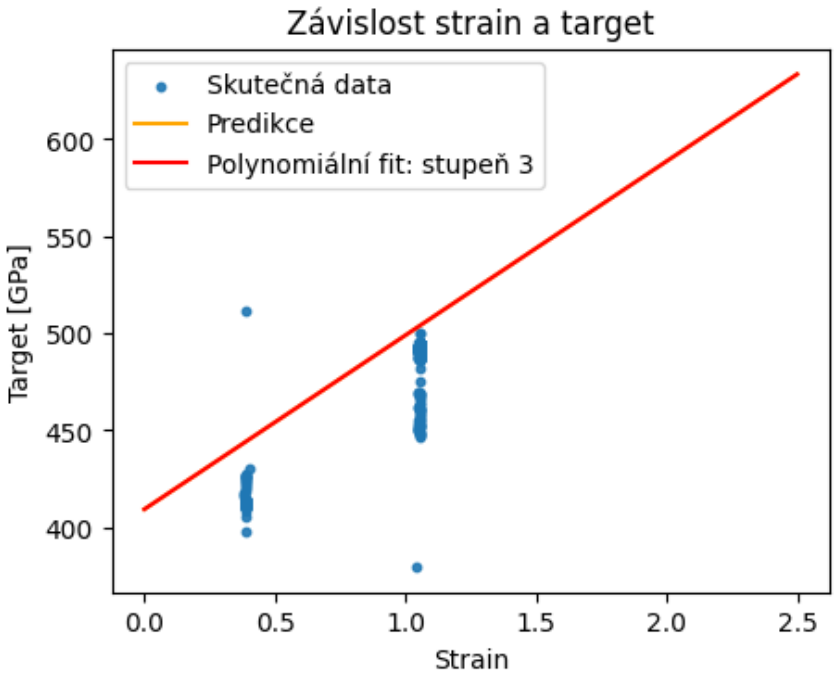
7.2 kNN s logaritmickou transformací



Obrázek 31: Aproximace predikce získané modelem kNN

MAPE predikce dosáhla přes všechna data hodnoty 4.36 %. Po polynomiální aproximaci došlo ke zhoršení na MAPE rovno 6.65 %. U této konkrétní slitiny došlo k prakticky téměř dokonalé predikci i aproximaci.

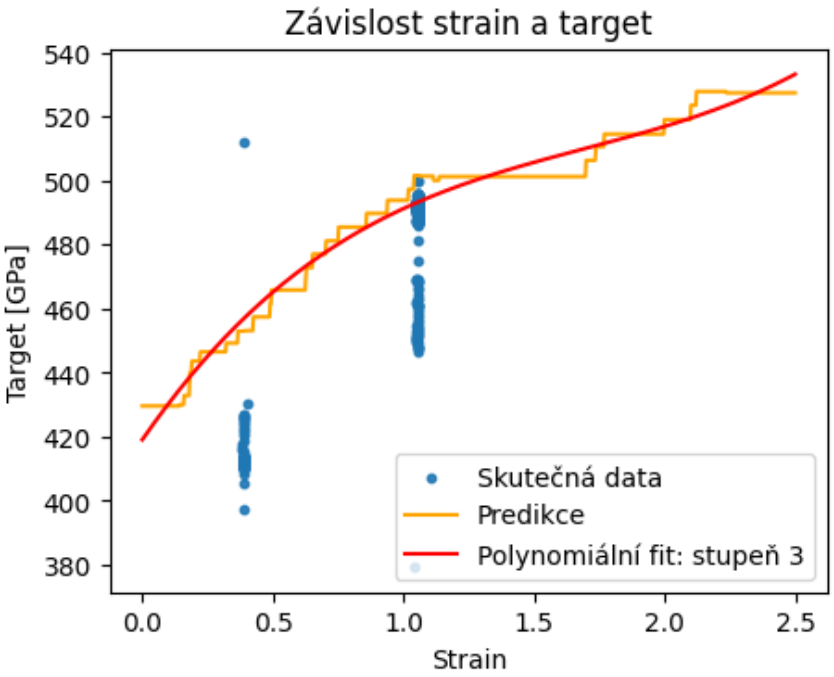
7.3 MLP s normalizací



Obrázek 32: Aproximace predikce získané modelem MLP

Zde je vidět, že aproximace ani predikce nedopadla u této slitiny příliš dobře, ale MAPE predikce přes všechna data dosáhla hodnoty 6.40 %. Polynomiální aproximací došlo pouze k minimálnímu zhoršení a to na hodnotu 6.45 %. Výsledek je tedy v závěru velmi blízko modelu kNN.

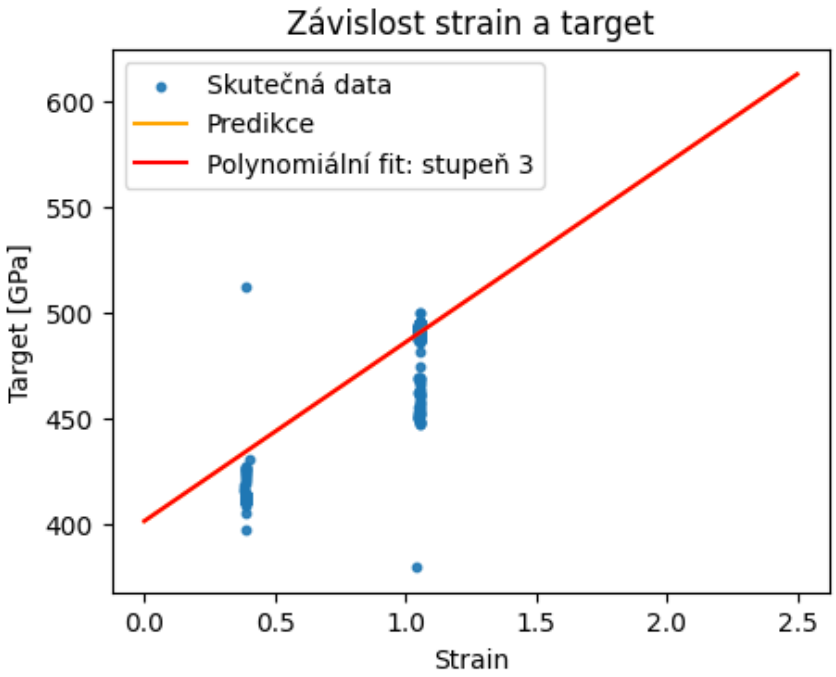
7.4 Extra Trees s Yeo-Johnson transformací



Obrázek 33: Aproximace predikce získané modelem Extra Trees

MAPE predikce tohoto modelu je 5.28 %. V tomto případě došlo u polynomiální aproximace ke zhoršení na hodnotu 7.31 %.

7.5 Lineární regrese se standardizací



Obrázek 34: Aproximace predikce získané modelem Lineární regrese

Není velkým překvapením že polynomiální fit sedí přesně na predikci a dosahuje tedy stejné odchylky jako samotná predikce a to MAPE 5.94 %.

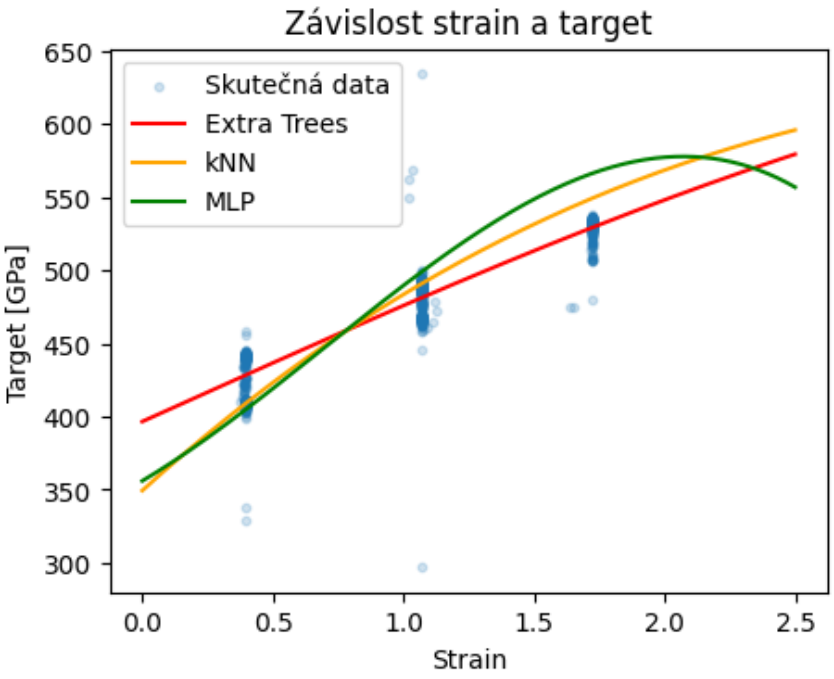
7.6 Testování na celém datasetu

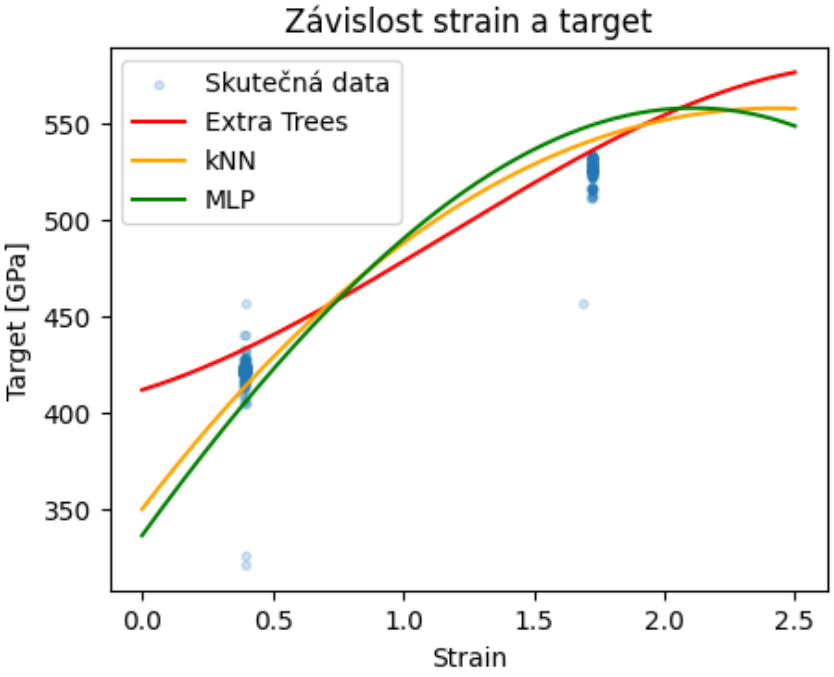
Nyní budou tyto modely testovány na původním celém datasetu. Tím se zjistí, jak optimalizované modely reagují na dosud neviděná data. A jak si poradí s větším objemem dat. Výsledky jsou zobrazeny v tabulce 6.

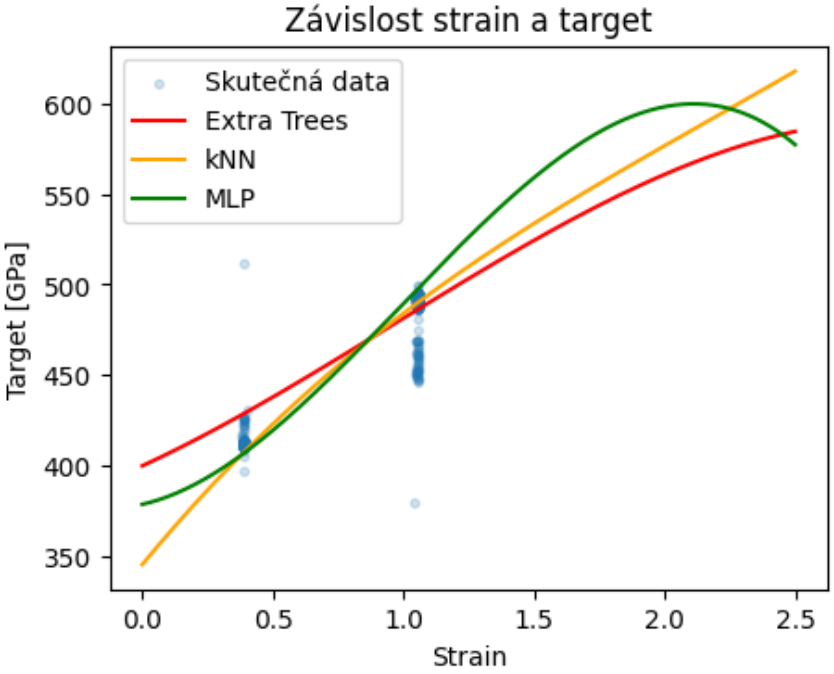
Model	**MAPE**	**MAPE aproximace**
kNN s logaritmickou transformací	3.75 %	**6.35 %**
MLP s normalizací	6.48 %	7.24 %
Extra Trees s Yeo-Johnson transformací	4.55 %	6.77 %
Lineární regrese se standardizací	9.45 %	9.45 %

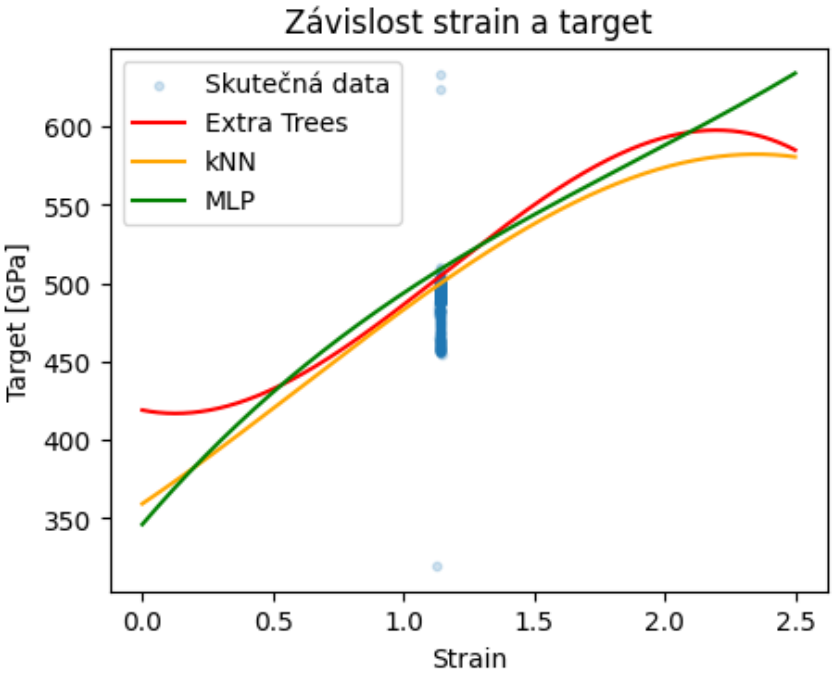
Tabulka 6: Srovnání výkonu modelů na celém datasetu

Grafické srovnání aproximací nejlepších modelů






Závislost strain a target
Skutečná data
Extra Trees
kNN
MLP
Target [GPa]
600
550
500
450
400
350
0.0
0.5
1.0
1.5
2.0
2.5
Strain


Závislost strain a target
Skutečná data
Extra Trees
kNN
MLP
Target [GPa]
600
550
500
450
400
350
0.0
0.5
1.0
1.5
2.0
2.5
Strain

7.7 Shrnutí výsledků

Jak bylo dokázáno na celém datasetu, tak nejlépe dopadl algoritmus kNN s logaritmickým předzpracováním dat, u kterého predikce dosáhla MAPE 3.75 % a následná aproximace této predikce polynomem třetího řádu MAPE 6.35 %. Dobrý výsledek také poskytl model Extra Trees s Yeo-Johnson transformací dat. Model lineární regrese ovšem komplexitu tohoto velkého datasetu nezvládl. Na zmenšeném datasetu, kde příznakový prostor není tak rozsáhlý tato metoda fungovala ale velice dobře. Velkou výhodou modelu lineární regrese je jeho rychlost, která byla ze všech modelů nejvyšší. Pořád velmi rychlý byl také model kNN. Výhodou obou těchto algoritmů je, že jejich výsledky nejsou ovlivněny žádnými počátečními podmínkami ani stochastickými prvky, což znamená, že při opakovaném spuštění a použití stejných dat vždy dosáhnou stejného výsledku.

Naopak model Extra Trees je závislý na počátečních podmínkách, což přináší určitou variabilitu výsledků. Tento model je také výrazně pomalejší (bez optimalizace) v porovnání s lineární regresí. Model MLP, který je nejpomalejší ze všech testovaných, také vykazuje značnou závislost na počátečních podmínkách a procesu trénování. Tyto charakteristiky však nejsou překvapivé vzhledem k složitosti a struktuře MLP. Tyto závislosti a výpočetní rychlosti můžou mít významný dopad, zejména při práci s rozsáhlými datovými sadami.

7.8 Možná rozšíření

Pro zvýšení robustnosti a přesnosti prediktivního modelu by mohlo být provedeno testování ještě širší škály algoritmů.

Kombinace více modelů do ensemble modelu, jako je bagging, boosting, stacking nebo forma váženého průměru predikcí, by mohla vést k zvýšení přesnosti a stability predikcí.

Otestování modelů na větším a různorodějším datasetu by mohlo poskytnout hlubší porozumění dynamikám procesu válcování a zvýšit obecnost modelu.

Vytvoření metriky, která by určovala, jak velkou věrohodnost můžeme jednotlivým predikcím přiřadit.

8 Závěr

Tato diplomová práce představila komplexní přístup k predikci síly nutné pro válcování plechů, což je klíčový aspekt ve výrobním procesu kovových materiálů. Práce začíná úvodem do problematiky, který nastiňuje základní motivaci a cíle výzkumu.

Teoretická část se věnuje představení válcování za studena a za tepla, popisuje celý proces válcování od výběru a přípravy materiálu, přes samotné válcování, až po normalizaci, chlazení a finální úpravy. Spolupráce s firmou PTSW poskytla potřebná data a reálný kontext pro aplikaci prediktivních modelů.

Byly představeny základy strojového učení, jeho klíčové kategorie, cíle a nezbytné podmínky pro jeho efektivní využití. Byl také ukázán význam předzpracování dat, který je nezbytný pro dosažení kvalitních výsledků, a přehled základních metod, které se v tomto procesu používají.

Praktická část se zaměřila na popis dat, definici úlohy, použití metod datové analýzy a vyhodnocení výsledků. V rámci analýzy byly využity různé algoritmy, včetně lineární regrese, kNN a stromových metod, s hodnocením jejich výkonu pomocí metriky MAPE.

Na základě požadavku firmy PTSW na predikce ve formě polynomu, byly výstupy regresních modelů aproximovány polynomem třetího řádu a následně testovány na celém datasetu. Z tohoto testování vyplynulo, že model lineární regrese není pro rozsáhlý dataset ideální volbou. Naopak model kNN s logaritmickou transformací dat dosáhl nejlepších výsledků s MAPE 6.35 %. Tento model se také ukázal jako vhodný, protože jeho přesnost se zvyšuje s množstvím dostupných dat.

Výsledky této práce mohou být použity v průmyslové praxi. Nebo případně mohou sloužit jako základ pro další výzkum.

9 Seznam použité literatury

[1] Ray, S. (2016, April 5). Introduction to Rolling Process. Cambridge University Press eBooks. https://doi.org/10.1017/cbo9781139879293.002

[2] Cold Rolling – Process Overview - Matmatch. (n.d.). Matmatch. https://matmatch.com/learn/process/cold-rolling

[3] Thorat, S. (2020, February 23). What is Hot Rolling - Advantages and Disadvantages. Mechanical Engineering Blog. https://learnmech.com/what-is-hot-rolling-advantages-and-disadvantages/

[4] PTSW. (n.d.). https://www.ptsw.cz/cs/

[5] Machine learning, explained | MIT Sloan. (2021, April 21). MIT Sloan. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[6] Data, A. (2023, November 14). Data Preprocessing: Steps, Techniques, and Importance in Machine Learning. https://blog.arkondata.com/data-preprocessing-in-machine-learning

[7] Brownlee J. (2020, August 18). Linear Discriminant Analysis for Dimensionality Reduction in Python. https://machinelearningmastery.com/linear-discriminant-analysis-for-dimensionality-reduction-in-python/

[8] Galli, S. (2023, November 5). Variance stabilizing transformations in machine learning. Train in Data's Blog. https://www.blog.trainindata.com/variance-stabilizing-transformations-in-machine-learning/

[9] Brownlee, J. (2019, August 8). How to Transform Data to Better Fit The Normal Distribution. MachineLearningMastery.com. https://machinelearningmastery.com/how-to-transform-data-to-fit-the-normal-distribution/

[10] Baheti, P. (2024, April 10). Train Test Validation Split: How To & Best Practices [2023]. V7. https://www.v7labs.com/blog/train-validation-test-set#h2

[11] Bayesian Hyperparameter Optimization: Basics & Quick Tutorial. (n.d.). https://www.run.ai/guides/hyperparameter-tuning/bayesian-hyperparameter-optimization

[12] What Is Supervised Learning? | IBM. (n.d.). https://www.ibm.com/topics/supervised-learning

[13] M. (2024, January 21). Unsupervised learning: principle and use. Data Science Courses | DataScientest. https://datascientest.com/en/unsupervised-learning-principle-and-use

[14] Hashemi-Pour, C., & Carew, J. M. (2023, August 16). reinforcement learning. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning

[15] Li, Y., Wang, H., Fan, J., & Geng, Y. (2022, December 27). A novel Q-learning algorithm based on improved whale optimization algorithm for path planning. PloS One. https://doi.org/10.1371/journal.pone.0279438

[16] Introduction to RL and Deep Q Networks. (n.d.). TensorFlow. https://www.tensorflow.org/agents/tutorials/0_intro_rl

[17] Team, G. (2024, May 10). Decision Trees Advantages and Disadvantages - Gyansetu. Gyansetu. https://www.gyansetu.in/blog/decision-trees-advantages-and-disadvantages/

[18] Glen, S. (2019, July 28). Decision Tree vs Random Forest vs Gradient Boosting Machines: Explained Simply. Data Science Central. https://www.datasciencecentral.com/decision-tree-vs-random-forest-vs-boosted-trees-explained/

[19] Bento, C. (2022, January 5). Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis. Medium. https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

[20] R. (2020, January 30). Principal component analysis (PCA): Explained and implemented. Medium. https://medium.com/@raghavan99o/principal-component-analysis-pca-explained-and-implemented-eeab7cb73b72

[21] Interpretace naměřených dat – Lean Six Sigma. (n.d.). https://lean6sigma.cz/interpretace-namerenych-dat/

[22] Kumar, A. (2022, April 16). Correlation Concepts, Matrix & Heatmap using Seaborn - Analytics Yogi. Analytics Yogi. https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/#google_vignette

[23] Kumar, V. (2022, February 6). Cross Validation in Sklearn | Hold Out Approach | K-Fold Cross Validation | LOOCV - MLK - Machine Learning Knowledge. MLK - Machine Learning Knowledge. https://machinelearningknowledge.ai/cross-validation-in-sklearn-hold-out-approach-k-fold-cross-validation-loocv/?utm_content=cmp-true

10 Apendix: holdout

Bez transformace

Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.38
GradientBoostingRegressor	1.94 %	89.05
XGBRegressor	**1.41 %**	**70.64**
RandomForestRegressor	1.52 %	82.45
DecisionTreeRegressor	1.69 %	108.63
ExtraTreesRegressor	1.59 %	94.82
MLPRegressor	3.11 %	154.33
LinearRegression	4.69 %	306.73

Inverzní transformace

Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.28
GradientBoostingRegressor	1.94 %	88.97
XGBRegressor	**1.40 %**	**68.43**
RandomForestRegressor	1.52 %	83.23
DecisionTreeRegressor	1.69 %	108.75
ExtraTreesRegressor	1.59 %	95.16
MLPRegressor	3.51 %	189.95
LinearRegression	5.88 %	621.28

Yeo-Johnson transformace

Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.28
GradientBoostingRegressor	1.94 %	89.05
XGBRegressor	**1.40 %**	**70.30**
RandomForestRegressor	1.53 %	83.55
DecisionTreeRegressor	1.69 %	109.03
ExtraTreesRegressor	1.59 %	94.50
MLPRegressor	1.94 %	93.58
LinearRegression	5.04 %	396.06

Logaritmická transformace

Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.30
GradientBoostingRegressor	1.94 %	89.09
XGBRegressor	**1.41 %**	**70.58**
RandomForestRegressor	1.53 %	83.96
DecisionTreeRegressor	1.69 %	107.61
ExtraTreesRegressor	1.59 %	94.20
MLPRegressor	2.55 %	119.98
LinearRegression	4.90 %	343.47

Standardizace

Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.28
GradientBoostingRegressor	1.94 %	89.05
XGBRegressor	**1.40 %**	**70.61**
RandomForestRegressor	1.53 %	82.89
DecisionTreeRegressor	1.69 %	107.99
ExtraTreesRegressor	1.58 %	93.81
MLPRegressor	2.08 %	95.25
LinearRegression	4.69 %	306.73

Normalizace

Metoda	MAPE	MSE
KNeighborsRegressor	1.45 %	74.38
GradientBoostingRegressor	2.00 %	90.62
XGBRegressor	**1.40 %**	**70.38**
RandomForestRegressor	1.42 %	71.82
DecisionTreeRegressor	1.46 %	80.25
ExtraTreesRegressor	1.47 %	78.97
MLPRegressor	4.88 %	337.76
LinearRegression	4.69 %	306.73

11 Apendix: k-fold

Bez transformace

Metoda	MAPE	MSE
KNeighborsRegressor	6.37 %	1084.16
GradientBoostingRegressor	5.48 %	711.04
XGBRegressor	7.32 %	1241.37
RandomForestRegressor	7.04 %	1406.43
DecisionTreeRegressor	9.37 %	2632.99
ExtraTreesRegressor	**5.16 %**	863.05
MLPRegressor	6.63 %	817.51
LinearRegression	5.94 %	**464.09**

Inverzní transformace

Metoda	MAPE	MSE
KNeighborsRegressor	9.23 %	2030.50
GradientBoostingRegressor	**5.37 %**	**694.37**
XGBRegressor	7.96 %	1464.79
RandomForestRegressor	6.90 %	1385.97
DecisionTreeRegressor	8.24 %	2096.26
ExtraTreesRegressor	5.68 %	1008.51
MLPRegressor	23.46 %	9514.93
LinearRegression	7.78 %	810.85

Yeo-Johnson transformace

Metoda	MAPE	MSE
KNeighborsRegressor	5.98 %	1052.79
GradientBoostingRegressor	5.50 %	712.85
XGBRegressor	6.86 %	1137.35
RandomForestRegressor	6.76 %	1274.93
DecisionTreeRegressor	8.60 %	2033.31
ExtraTreesRegressor	**5.05 %**	844.11
MLPRegressor	6.06 %	**541.31**
LinearRegression	6.30 %	571.52

Logaritmická transformace

Metoda	MAPE	MSE
KNeighborsRegressor	5.17 %	758.63
GradientBoostingRegressor	5.54 %	734.99
XGBRegressor	6.95 %	1145.85
RandomForestRegressor	7.11 %	1451.64
DecisionTreeRegressor	8.95 %	2667.84
ExtraTreesRegressor	**5.15 %**	862.73
MLPRegressor	6.13 %	**469.44**
LinearRegression	5.98 %	475.19

Standardizace

Metoda	MAPE	MSE
KNeighborsRegressor	6.37 %	1084.16
GradientBoostingRegressor	5.48 %	711.04
XGBRegressor	7.32 %	1241.37
RandomForestRegressor	7.04 %	1406.43
DecisionTreeRegressor	9.37 %	2632.99
ExtraTreesRegressor	**5.16 %**	863.05
MLPRegressor	6.63 %	817.51
LinearRegression	5.94 %	**464.09**

Normalizace

Metoda	MAPE	MSE
KNeighborsRegressor	20.65 %	8885.90
GradientBoostingRegressor	14.04 %	4160.86
XGBRegressor	20.18 %	14963.96
RandomForestRegressor	12.59 %	3484.49
DecisionTreeRegressor	12.18 %	3029.67
ExtraTreesRegressor	11.72 %	3777.98
MLPRegressor	**5.92 %**	**499.07**
LinearRegression	73.56 %	111218.41