Regresia pomocou metódy podporných vektorov: Nástroj pre presné a robustné predikcie

Support Vector Regression: A tool for accurate and robust predictions

Andrej Bednařík[1]

Abstrakt
Tento príspevok je zameraný na Support Vector Regression (SVR), pokročilú metodológiu strojového učenia pre riešenie regresných problémov v rôznych aplikáciách. SVR vychádza z algoritmov Support Vector Machine, využíva podporné vektory na modelovanie prediktívnych funkcií, ktoré minimalizujú chyby predikcie v rámci preddefinovaného prahu. Tento robustný mechanizmus umožňuje vysokú presnosť aj pri komplexných a šumivých dátových súboroch. Príspevok rieši princípy, metódy a aplikácie SVR, pričom zdôrazňuje prispôsobivosť na nelineárne problémy prostredníctvom kernelových metód a využitie.

Kľúčové slová
Support Vector Regression (SVR), Kernel, Python, Predikcia

Abstract
This document explores Support Vector Regression (SVR), an advanced machine learning methodology for solving regression problems across various applications. Originating from Support Vector Machine algorithms, SVR utilizes support vectors to model predictive functions that minimize prediction errors within a predefined threshold. This robust mechanism allows for high accuracy even with complex and noisy data sets. The paper discusses the principles, methodologies, and applications of SVR, emphasizing its adaptability to nonlinear problems through kernel methods and its applications.

Key words
Support Vector Regression (SVR), Kernel, Python, Prediction

JEL classification
C61, C89

1. Úvod

Support Vector Regression (SVR) predstavuje sofistikovanú metodiku v rámci strojového učenia, navrhnutú na efektívne riešenie regresných problémov v širokom spektre aplikácií. Tento prístup vychádza z algoritmu Support Vector Machine (SVM), ktorý bol pôvodne vyvinutý pre účely klasifikácie. Princíp SVM spočíva v identifikácii a optimalizácii rozhodovacích hraníc medzi jednotlivými klasifikačnými triedami, pričom SVR tento koncept rozširuje do domény regresnej analýzy (Vapnik, 1995). V kontexte SVR sa využíva koncept podporných vektorov na modelovanie prediktívnej funkcie, ktorá sa snaží minimalizovať chyby predikcie tak, že odchýlky medzi predpovedanými a skutočnými hodnotami sú držané v rámci vopred definovaného prahu známeho ako ε (epsilon). Tento mechanizmus umožňuje SVR zachovať vysokú presnosť predikcie aj v prípade komplexných a „šumivých“ dátových súborov, čo je neoceniteľné v mnohých praktických aplikáciách. Hyperrovina v modeli SVR je

[1] Ing. Andrej Bednařík, Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra matematiky a aktuárstva, Dolnozemská cesta 1, 852 35 Bratislava, andrej.bednarik@euba.sk

reprezentovaná pomocou vybraných trénovacích bodov, známych ako podporné vektory. Tieto body sú kritické pre model, pretože určujú hranice predikčného intervalu a sú priamo zapojené do výpočtu konečného rozhodovacieho modelu (Smola & Schölkopf, 2004). SVR má svoje korene v 90. rokoch 20. storočia, keď boli položené základy teórie strojového učenia v práci Vapnika a jeho kolegov. Tieto algoritmy sú postavené na pevnom základe teórie štatistického učenia a konceptu štrukturálneho rizika, ktoré cieľavedome minimalizuje pravdepodobnosť chyby na nevidených dátach a zároveň redukuje riziko preučenia modelu (Vapnik, 1995). Dnes sa SVR aplikuje v širokom spektre odvetví, od predpovedí na finančných trhoch až po optimalizáciu energetických systémov a vývoj pokročilých zdravotníckych diagnostických nástrojov. Výhody SVR, ako sú robustnosť, presnosť a schopnosť efektívne spracovávať veľké objemy dát, sú zásadné pre tieto aplikácie.

2. Princípy a metódy SVR

Základný princíp SVR je že pracuje na princípe štruktúrneho minimalizovania rizika, ktorý je základom teórie strojového učenia Vapnika (Vapnik, 1995). Tento prístup sa snaží nájsť rovnováhu medzi zložitosťou modelu a mierou, do akej sa model prispôsobuje trénovacím dátam, aby sa minimalizovala chyba na nevidených dátach.

Loss funkcia a ε-insensitivity: SVR zavádza koncept ε-insenzitívnej loss funkcie, ktorá ignoruje chyby v predikcii, ktoré sú menšie ako ε. Tento prístup umožňuje určitú mieru odchýlok bez toho, aby boli penalizované, čo pomáha predchádzať preučeniu modelu (Smola & Schölkopf, 2004).

Optimalizačný problém: Cieľom SVR je nájsť funkciu, ktorá najlepšie oddelí všetky dáta plus a mínus ε od skutočnej cieľovej hodnoty y. To sa dosahuje riešením optimalizačného problému, kde sa minimalizuje norma váhového vektora w a zároveň sa trestajú odchýlky, ktoré sú väčšie ako ε (Bishop, 2006).

Kernelová metóda: Podobne ako SVM, aj SVR môže využívať kernelovú metódu. Kernelová metóda využíva matematické funkcie, nazývané kernelové funkcie, na transformáciu pôvodného vstupného priestoru do nového, typicky vyššieho dimenzionálneho priestoru, čo umožňuje efektívne riešenie nelineárnych regresných problémov. kde sú vzťahy medzi dátovými bodmi jednoduchšie alebo dokonca lineárne separovateľné. Tento prístup umožňuje SVM a SVR efektívne pracovať s komplexnými alebo nelineárnymi vzťahmi bez explicitného zvyšovania dimenzií vstupných dát, čo by bolo výpočtovo veľmi náročné. Bežné kernely zahŕňajú lineárne, polynomiálne, radiálne bázové funkcie (RBF) a sigmoidálne kernely (Hofmann, Schölkopf, & Smola, 2008).

2.1 ε-insensitivity loss function a optimalizačný problém

V kontexte Support Vector Regression (SVR) je kľúčovou súčasťou modelu tzv. ε-insenzitívna loss funkcia, ktorá hrá dôležitú úlohu v znižovaní preučenia a zvyšovaní schopnosti generalizácie modelu (Smola & Schölkopfom, 2004).
Vapnikova ε-insenzitívna loss funkcia, označovaná ako L_ε, je definovaná nasledovne:

$$L_\varepsilon\big(y, f(x)\big) = \max(0, |y - f(x)| - \varepsilon), \tag{1}$$

kde y je skutočná hodnota cieľovej premennej, $f(x)$ je predikovaná hodnota modelom, ε(epsilon) je nenulová hranica, ktorá definuje prah, do ktorého sú chyby ignorované.

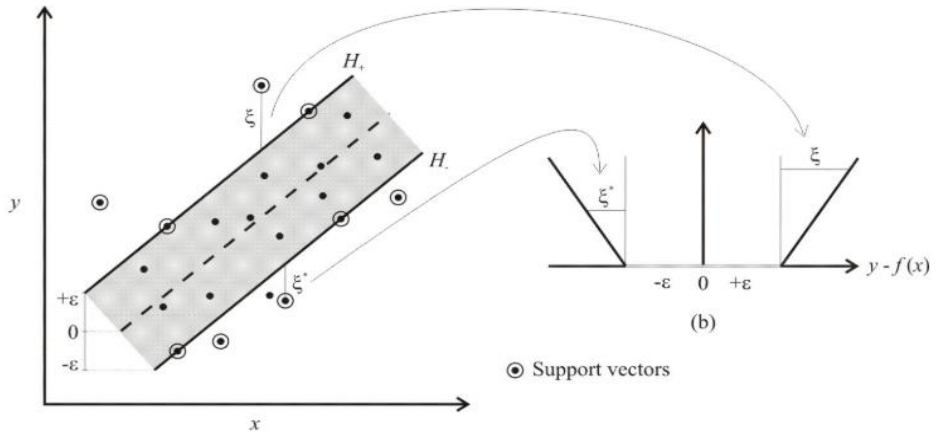
Funkcia L_ε vracia hodnotu 0 pre všetky chyby predikcie $|y - f(x)|$, ktoré sú menšie alebo rovné ε. Toto znamená, že chyby, ktoré sú v rámci prahu ε, nepripisujú žiadnu stratu, a teda model nie je penalizovaný za tieto malé odchýlky. Ak je odchýlka väčšia ako ε, potom je strata počítaná ako rozdiel medzi absolútnou hodnotou chyby a ε. Hlavnou výhodou tohto

prístupu je jeho schopnosť minimalizovať vplyv šumu a náhodných fluktuácií v trénovacích dátach, čo zvyšuje robustnosť a spoľahlivosť modelu. Funkcia umožňuje modelu SVR presne predpovedať dôležité vzory v dátach bez nadmernej citlivosti na malé odchýlky. Teda Vapnikova lineárna ε-insenzitívna stratová funkcia definuje "trúbka" s polomerom ε okolo cieľových hodnôt y. Potom platí:

$$|y - f(x)| - \varepsilon = \xi, \text{ pre dátové body "nad" trubkou.} \quad (2)$$

$$|y - f(x)| - \varepsilon = \xi^*, \text{ pre dátové body "pod" trubkou.} \quad (3)$$

Obr. 1: Znázornenie slack premenných



Zdroj: (Lins et al., 2010)

2.2 Kernel

Kernel, známy tiež ako jadro, je fundamentálna súčasť mnohých metód strojového učenia, najmä v kontexte Support Vector Machines (SVM) a príbuzných techník, ako je Support Vector Regression (SVR). Kernelové funkcie transformujú pôvodné vstupné dáta do vyššieho dimenzionálneho priestoru, umožňujúc efektívne riešenie nelineárnych problémov bez nutnosti explicitne zvyšovať dimenzionalitu dát. Tento proces je známy ako "kernelový trik" a je zásadný pre manipuláciu s komplexnými vzťahmi medzi dátami v nelineárnych priestoroch (Shawe-Taylor & Cristianini, 2004).

Typy kernelov a ich aplikácie:

Lineárny kernel s kernelovou fukniou: $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$.

- Lineárny kernel je najjednoduchší a počíta skalárny súčin dvoch vektorov. Je účinný, keď sú dáta lineárne separovateľné a nevyžadujú transformáciu do vyššieho dimenzionálneho priestoru.

Polynomiálny kernel s kernelovou funkciou: $\mathrm{K}(\mathbf{x}, \mathbf{y}) = (\gamma \cdot \langle \mathbf{x}, \mathbf{y} \rangle + r)^d$.

- Polynomiálny kernel umožňuje SVM modelovať rozhodovacie hranice v tvare polynómov určitého stupňa d. Jeho flexibilita pri modelovaní nelineárnych vzťahov je významnou výhodou v aplikáciách, kde lineárne modely zlyhávajú (Chang & Lin, 2011).

RBF kernel s funkciou: $K(\mathbf{x}, \mathbf{y}) = exp(-\gamma \parallel \mathbf{x} - \mathbf{y} \parallel^2)$.

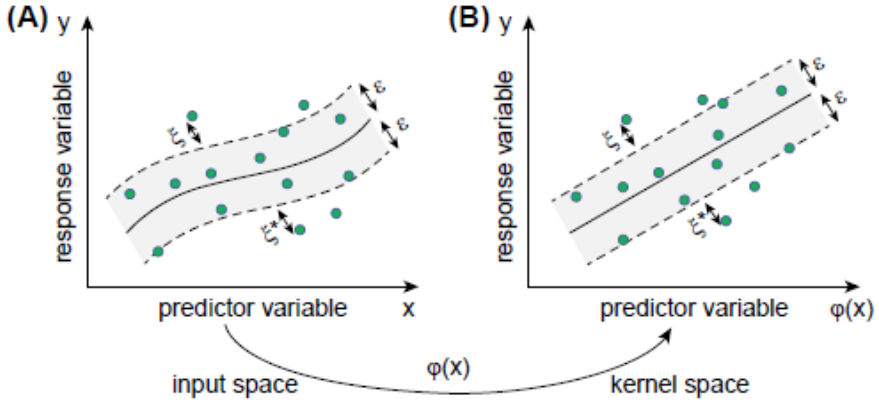
- RBF kernel, často nazývaný aj Gaussovský kernel, je veľmi obľúbený vďaka svojej schopnosti efektívne mapovať prvky do nekonečne dimenzionálneho priestoru, čo je ideálne pre veľmi zložité vzťahy medzi dátovými bodmi.

Sigmoidálny kernel s funkciou: $K(\mathbf{x}, \mathbf{y}) = tanh(\gamma \cdot \langle \mathbf{x}, \mathbf{y} \rangle + r)$.

- Sigmoidálny kernel, inšpirovaný neurónovými aktiváciami, môže poskytnúť výstupy, ktoré pripomínajú aktivačné funkcie používané v neurónových sieťach, čo je užitočné pre určité typy klasifikačných problémov.

Duálna formulácia tohto optimalizačného problému zjednodušuje výpočty a umožňuje využitie kernelových funkcií na zvládnutie nelineárnych vzťahov v dátach. Duálny problém využíva Lagrangeove multiplikátory na preformulovanie optimalizačného problému, ktorý sa následne rieši pomocou kvadratického programovania. Voľba správneho kernelu závisí od povahy dát a špecifických požiadaviek problému. Správne nastavenie kernelu môže dramaticky zvýšiť výkonnosť modelu, zatiaľ čo nesprávna voľba môže viesť k nedostatočnému učeniu alebo preučeniu (Hofmann, Schölkopf, & Smola, 2008).

Obr. 2: Transformácia priestoru



Zdroj: (Zhang & O'Donnell, 2020)

Na obrázku 2 je grafické znázornenie nelineárneho ε-SVR. Funkcia mapovania φ sa používa na transformáciu dát z vstupného priestoru (A), kde nie je možné lineárne oddeliť dáta, do vyššie-dimenzionálneho kernelového priestoru (B), kde môžu byť dáta oddelené lineárnou hyperrovinou.

3. Lineárny ε – SVR model

Cieľom ε-SVR (ε-Support Vector Regression) je odhadnúť funkciu s obmedzením, že odhad každého vstupného dátového bodu má najviac ε odchýlku od svojej skutočnej hodnoty odpovede, vytvorením ε-insenzitívnej trubice symetricky okolo odhadnutej funkcie. Matematická formulácia lineárneho ε-SVR môže byť vyjadrená nasledovne. Predpokladajme, že máme súbor trénovacích dát $\{(x_{11}..... x_{1k}, y_1), ..., (x_{n1} ..... x_{nk}, y_n)\}$, kde x_{i1},...., x_{ik}, sú hodnoty vstupných dát a y_i je cieľový výstup, $i=1,...,n$. Prípad lineárnej regresnej funkcie f má tvar:

$$y = f(x) = \langle \boldsymbol{\omega}, \mathbf{x} \rangle + b, \tag{4}$$

kde $\langle \boldsymbol{\omega}, \mathbf{x} \rangle$ označuje skalárny súčin vektora vstupných dát (vysvetľujúce premenné) $\mathbf{x}$ a vektor váh $\boldsymbol{\omega}$ a b je konštanta ktorá nie je pevne daná, ale je skôr parameter, ktorý sa optimalizuje počas trénovania modelu. Optimalizácia b sa vykonáva spolu s optimalizáciou vektora váh ω s cieľom minimalizovať chybu predikcie. V ε-SVR sa aproximácia funkcie f vykonáva nájdením ε-insenzitívnej trubice, ktorá je čo najplochejšia, čo je formálne označované ako plochosť.

Viacúčelový optimalizačný problém SVR možno zapísať nasledovne

$$min\left\{\frac{1}{2}||\boldsymbol{\omega}||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)\right\} \tag{5}$$

Pričom platia nasledovné obmedzenia:

$$\begin{gathered} y_i - \langle \boldsymbol{\omega}, \mathbf{x_i} \rangle - b \leq \varepsilon + \xi_i \\ \langle \boldsymbol{\omega}, \mathbf{x_i} \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \end{gathered} \tag{6}$$

kde $\mathbf{x_i}$ reprezentuje vstupné prvky, kde $\boldsymbol{\omega}$ obsahuje váhy priradené ku každému vstupnému prvku (feature) vo vstupnom vektore $\mathbf{x_i}$,tieto váhy určujú, aký vplyv bude mať každý vstupný prvok na výstupnú hodnotu modelu a C je regulačný parameter, ktorý kontroluje rovnováhu medzi hladkosťou modelu a veľkosťou odchýlok. $||\boldsymbol{\omega}||^2$označuje kvadrát normy váhového vektora $\boldsymbol{\omega}$ (Zhang & O'Donnell, 2020).

Optimalizácia v SVR zahŕňa minimalizáciu kombinácie normy váhového vektora a sumy ε-insenzitívnych strát cez všetky trénovacie dáta. Základný matematický model SVR sa snaží nájsť funkciu, ktorá minimalizuje chybu predikcie v rámci prahu ε a zároveň udržiava hladkosť modelu. Termín "hladkosť modelu" je kľúčový pre schopnosť modelu generalizovať, čo znižuje riziko preučenia. Hladký model efektívne predpovedá výsledky na nevidených dátach vďaka svojej jednoduchosti a odolnosti voči šumu v trénovacích dátach.
Definícia hladkosti modelu zahŕňa:

- Redukcia zložitosti: Hladký model má jednoduchšiu štruktúru, čo znamená menej parametrov alebo nižší stupeň polynómu. Toto pomáha zabezpečiť, že model nie je nadmerne prispôsobený na špecifické vzory alebo šum v trénovacích dátach.
- Generalizácia: Hladký model je menej citlivý na malé variácie v dátach, čo znižuje pravdepodobnosť, že model zachytí náhodné vzory, ktoré nie sú reprezentatívne pre všeobecné dáta.

Hladkosť modelu sa dosahuje pomocou

- Normy váhového vektora($||\boldsymbol{\omega}||^2$): Minimalizácia normy váhového vektora $\boldsymbol{\omega}$, ktorý predstavuje koeficienty regresného modelu, pomáha udržiavať model jednoduchý a hladký. Nižšie hodnoty normy znamenajú menej zložitý model, ktorý je všeobecne lepší pre generalizáciu.
- Penalizáciou odchýlok(ξ_i, ξ_i^*): V SVR sú zavedené takzvané slack premenné ξ_i, ξ_i^*, ktoré umožňujú určitú flexibilitu v prekročení prahu ε. Penalizácia týchto premenných zabezpečuje, že model neignoruje veľké odchýlky, čo pomáha vyvážiť medzi prísnosťou a prílišnou flexibilitou.

Táto rovnováha medzi udržaním modelu jednoduchým a zároveň dostatočne flexibilným na presné modelovanie dát, je kľúčom k úspešnému strojovému učeniu a predstavuje dôležitý aspekt pri návrhu a implementácii regresných modelov ako je SVR (Smola & Schölkopf, 2004). Slack premenné sú nevyhnutné pre správne fungovanie modelu, najmä pri riešení dát s prítomným šumom alebo výstupnými odchýlkami. Tieto premenné, známe ako ξ_i, ξ_i^*, umožňujú modelu tolerovať chyby predikcie, ktoré presahujú určený prah ε ,bez toho, aby boli príliš penalizované. Táto vlastnosť zaisťuje, že model je odolný voči preučeniu a zároveň zachováva jeho schopnosť generalizácie na nevidené dáta.

- $\boldsymbol{\xi_i}$ meria veľkosť odchýlky, keď predpovedaná hodnota presiahne skutočnú hodnotu o viac ako ε.
- $\boldsymbol{\xi_i^*}$ meria veľkosť odchýlky, keď skutočná hodnota presiahne predpovedanú o viac ako ε.

Tieto premenné sú penalizované v rámci optimalizačného problému SVR, čo zabezpečuje, že model dokáže spracovať dáta s inherentnými odchýlkami bez straty prediktívnej schopnosti. Flexibilita, ktorú slack premenné poskytujú, je kritická pre aplikácie v reálnom svete, kde dáta často obsahujú šum alebo sú neúplné (Zhu & Hastie, 2005).

4. Kernelový SVR model

Vyššie uvedená časť popisuje lineárny model ε-SVR, ktorý pracuje so vstupnými dátami v ich priestoroch znakov a predpokladá, že funkcia *f(x)* je lineárna funkcia. Aby sme umožnili ε-SVR spracovávať nelineárne dáta, môžeme zaviesť kernelovú funkciu, ktorá transformuje pôvodné vstupné dáta do vyššieho dimenzionálneho priestoru, nazývaného kernelový priestor. Používanie kernelov je jedným z najbežnejších prístupov v SVM (pre regresiu a klasifikáciu), pretože nie je potrebné riešiť vysoko-rozmernú separačnú hyperpovrch v vstupnom priestore, čo je oveľa komplikovanejšie v porovnaní s riešením lineárnej optimalizácie v kernelovom priestore.

Optimalizačný problém je často riešený v jeho duálnej forme, ktorá umožňuje využitie kernelových funkcií pre riešenie nelineárnych vzťahov v dátach. Duálna formulácia zahŕňa Lagrangeove multiplikátory. Štandardná metóda dualizácie využívajúca Lagrangeove multiplikátory je opísaná nasledovne:

$$\begin{aligned} \boldsymbol{L} = \frac{\mathbf{1}}{\mathbf{2}}||\boldsymbol{\omega}||^2 + \boldsymbol{C}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{\xi_i} + \boldsymbol{\xi_i^*}) - \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{\alpha_i}(\boldsymbol{\varepsilon} + \boldsymbol{\xi_i} - \boldsymbol{y_i} + \langle \boldsymbol{\omega}, \mathbf{x_i}\rangle + \boldsymbol{b}) \\ - \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{\alpha_i^*}(\boldsymbol{\varepsilon} + \boldsymbol{\xi_i^*} + \boldsymbol{y_i} - \langle \boldsymbol{\omega}, \mathbf{x_i}\rangle - \boldsymbol{b}) - \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{\eta_i}\boldsymbol{\xi_i} + \boldsymbol{\eta_i^*}\boldsymbol{\xi_i^*}) \end{aligned} \tag{7}$$

Duálne premenné v (7) musia spĺňať podmienky pozitívnosti, t.j. $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$. Zo sedlového bodu vyplýva, že parciálne derivácie L vzhľadom na primárne premenné $(\boldsymbol{\omega}, b, \xi_i, \xi_i^*)$ musia zmiznúť pre optimálnosť.

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i) = 0 \tag{8}$$

$$\frac{\partial L}{\partial \omega} = \boldsymbol{\omega} - \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)\mathbf{x_i} = 0 \tag{9}$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \tag{10}$$

Substituovaním (8), (9) a (10) do (7) vznikne duálny optimalizačný problém.

$$max\left\{-\frac{1}{2}\sum_{i,j=0}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle \mathbf{x_i}, \mathbf{x_j}\rangle - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n}y_i(\alpha_i - \alpha_i^*)\right\} \tag{11}$$

Za podmienok

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$$

$$\alpha_i, \alpha_i^* \in [0, C]$$

Duálne premenné η_i, η_i^* prostredníctvom podmienky (10) boli odstránené pre odvodenie (11). Rovnicu (9) možno prepísať takto:

$$\boldsymbol{\omega} = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\mathbf{x_i} \tag{12}$$

podľa vzťahu (12) potom

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\langle \mathbf{x_i}, \mathbf{x}\rangle + b \tag{13}$$

Výraz (12) je takzvaná expanzia podporných vektorov, t.j. **ω** môže byť úplne popísané ako lineárna kombinácia trénovacích vzorov $\mathbf{x_i}$. Algoritmus SV (Support Vector) môže byť nelineárny jednoduchým mapovaním trénovacích vzorov $\mathbf{x_i}$ do viacrozmerného priestoru pomocou kernelu $\varphi: X \rightarrow \Im$ a následným použitím štandardného algoritmu SV regresie. Expanzia v (12) potom vyzerá nasledovne (Basak et al., 2007):

$$\boldsymbol{\omega} = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\varphi(\mathbf{x_i}) \tag{14}$$

podľa vzťahu (14) potom

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)k(\mathbf{x_i}, \mathbf{x}) + b \tag{15}$$

kde α_i, α_i^* sú Lagrangeove multiplikátory. Kernelová funkcia $k(\mathbf{x_i}, \mathbf{x})$bola definovaná ako lineárny skalárny súčin nelineárneho zobrazenia, t.j.

$$k(\mathbf{x_i}, \mathbf{x}) = \varphi(\mathbf{x_i})\varphi(\mathbf{x}) \tag{16}$$

5. Implementácia SVR na dataset pomocou programovacieho jazyka Python

Na začiatku každého projektu z oblasti strojového učenia je kľúčové mať jasne definovaný cieľ. Keď presne vieme, čo chceme dosiahnuť, môžeme pristúpiť k zberu a prvotnej analýze dát. Následne je nevyhnutné dáta očistiť, čo zahŕňa odstránenie chýb, úpravu formátu a rozdelenie dát na tréningové, validačné a testovacie súbory. S takto pripravenými dátami môžeme vyberať vhodné modely na ich spracovanie. Po vytvorení a otestovaní modelu

nasleduje fáza jeho optimalizácie a prípadného nasadenia do praxe. Je dôležité sa nezastaviť len pri prvej verzii modelu, ale pravidelne ho aktualizovať a prispôsobovať novým podmienkam a poznatkom. V konečnej fáze je model hodnotený na testovacej množine, čo poskytuje informácie o jeho reálnej efektivite a pripravenosti na implementáciu. Proces strojového učenia je však často iteratívny, a preto je nevyhnutné model priebežne monitorovať, aktualizovať a prispôsobovať ho s ohľadom na nové požiadavky alebo zistené nedostatky.

5.1 Dataset expenses

Dataset "expenses.csv" obsahuje údaje o výške poistného a zdravotných charakteristikách jednotlivcov. Tento dataset zahŕňa informácie ako vek, pohlavie, Body Mass Index (BMI), počet detí, fajčiarske zvyky a geografickú oblasť, kde osoba žije. Taktiež obsahuje údaje o výške poistného nákladu, ktorý jednotlivci platia(ročne), na základe ktorých je možné skúmať vzťahy medzi týmito faktormi a výškou poistného**.**

Obr. 3:Pohlaď na premenné datasetu expenses

	age	sex	bmi	children	smoker	region	charges
0001	19	female	27.9	0	yes	southwest	16884.924
0002	18	male	33.77	1	no	southeast	1725.5523
0003	28	male	33	3	no	southeast	4449.462
0004	33	male	22.705	0	no	northwest	21984.47061
0005	32	male	28.88	0	no	northwest	3866.8552
0006	31	female	25.74	0	no	southeast	3756.6216
0007	46	female	33.44	1	no	southeast	8240.5896
0008	37	female	27.74	3	no	northwest	7281.5056

Zdroj: Vlastné spracovanie

Dataset teda obsahuje nasledujúce premenné:

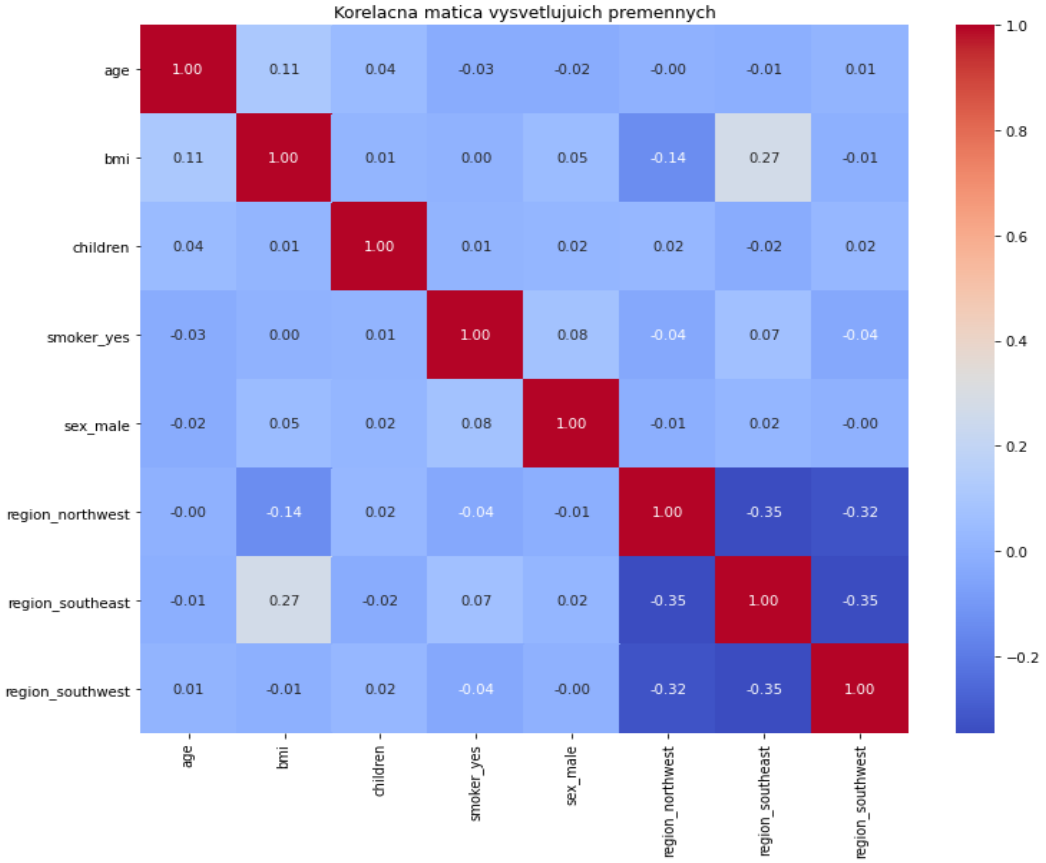
- **age (vek)**: Vek jednotlivca.
- **sex (pohlavie)**: Pohlavie jednotlivca (muž alebo žena).
- **bmi**: Index telesnej hmotnosti, ktorý je meradlom telesného tuku na základe výšky a váhy.
- **children (deti)**: Počet detí/závislých osôb, ktoré sú pokryté zdravotným poistením.
- **smoker (fajčiar)**: Udáva, či je jednotlivec fajčiar (áno alebo nie).
- **region (región)**: Región bydliska jednotlivca v Spojených štátoch (severovýchod, severozápad, juhovýchod, juhozápad).
- **charges (poplatky)**: Poplatky za zdravotné poistenie fakturované jednotlivcovi.

Zobrazené hodnoty v korelačnej matici na obrázku 4 sú hodnoty Pearsonovho korelačného koeficientu. Táto korelačná matica vysvetľujúcich premenných v datasete odhaľuje, že medzi väčšinou premenných sú veľmi slabé alebo žiadne korelácie, čo naznačuje nízku úroveň multikolinearity, čo je priaznivé pre modelovanie. Celkovo táto analýza naznačuje, že premenné sú vhodné na použitie v regresných modeloch, pretože nízka multikolinearita by nemala viesť k nestabilite modelu.

Blok kódu na obrázku 5 importuje potrebné knižnice a moduly, ktoré sa používajú na spracovanie dát, modelovanie a vizualizáciu v projekte.

Blok kódu na obrázku 6 načíta dataset zo súboru expenses.csv, vykoná one-hot encoding pre kategorizované premenné smoker, sex a region, rozdelí dáta na vstupné premenné `X` a cieľovú premennú `y`, a následne rozdelí dáta na tréningovú a testovaciu množinu v pomere 80:20. Tento pripravený dataset je potom pripravený na použitie v rôznych modeloch strojového učenia.

Obr. 4: Korelačná matica vysvetľujúcich premenných



Zdroj: Vlastné spracovanie

Obr. 5: Použité knižnice

```
import pandas as pd
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import seaborn as sns
```

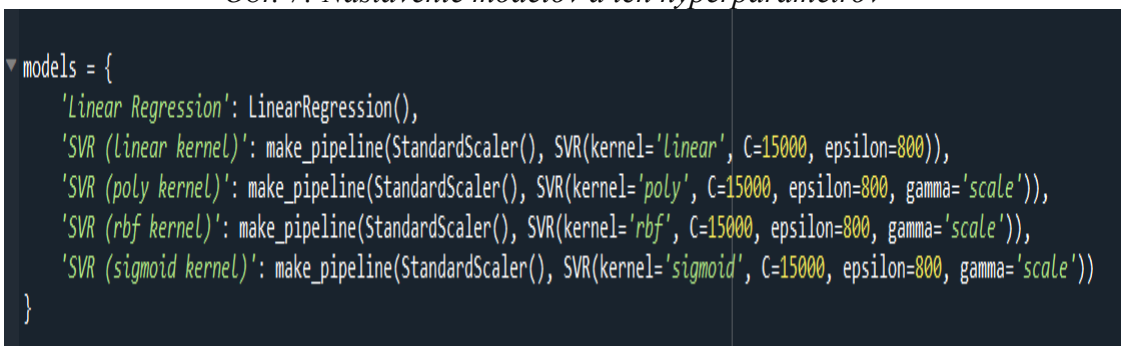
Zdroj: Vlastné spracovanie

Obr. 6: Práca s datasetom

```
data_path = 'C:\\Users\\PC\\Desktop\\expenses.csv'
hr_data = pd.read_csv(data_path)
hr_data = pd.get_dummies(hr_data, columns=['smoker', 'sex', 'region'], drop_first=True)
X = hr_data.drop('charges', axis=1)
y = hr_data['charges']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Zdroj: Vlastné spracovanie

Obr. 7: Nastavenie modelov a ich hyperparametrov



```
models = {
    'Linear Regression': LinearRegression(),
    'SVR (linear kernel)': make_pipeline(StandardScaler(), SVR(kernel='linear', C=15000, epsilon=800)),
    'SVR (poly kernel)': make_pipeline(StandardScaler(), SVR(kernel='poly', C=15000, epsilon=800, gamma='scale')),
    'SVR (rbf kernel)': make_pipeline(StandardScaler(), SVR(kernel='rbf', C=15000, epsilon=800, gamma='scale')),
    'SVR (sigmoid kernel)': make_pipeline(StandardScaler(), SVR(kernel='sigmoid', C=15000, epsilon=800, gamma='scale'))
}
```

Zdroj: Vlastné spracovanie

Blok kódu na obrázku 7 definuje slovník modelov s rôznymi typmi regresných modelov, vrátane lineárnej regresie a SVR s rôznymi kernelmi. Každý SVR model je obalený v `pipeline`, ktorá zahŕňa štandardizáciu vstupných dát pomocou `StandardScaler` a aplikáciu SVR s konkrétnymi hyperparametrami (pre dataset expenses sa po manuálnom testovaní hyperparametre nastavené na C=15000, ε=800, γ='scale' ukázali ako najefektívnejšie). Každý dataset má svoje vlastné charakteristiky, ako je štruktúra, množstvo šumu a komplexnosť vzťahov medzi premennými, čo ovplyvňuje optimálne nastavenie hyperparametrov.

Hyperparametre sú parametre, ktoré výrazne ovplyvňujú výkon a schopnosť modelu generalizovať. V kontexte Support Vector Regression (SVR) s rôznymi kernelmi sú hlavnými hyperparametrami C, epsilon (ε) a gamma (γ). Tu je vysvetlenie, ako tieto hyperparametre ovplyvňujú model:

Hyperparameter C (Regularizačný parameter):

- Funkcia**:** C určuje, ako veľmi chce model minimalizovať chyby. Vyvažuje medzi dvoma cieľmi: minimalizovať chyby na tréningovej množine a udržať model jednoduchý (vyhnúť sa overfittingu).
- Vplyv:
    - Vysoké hodnoty C**:** Model sa snaží minimalizovať chyby na tréningových dátach, čo môže viesť k overfittingu.
    - Nízke hodnoty C: Model je viac penalizovaný za chyby a môže mať tendenciu byť jednoduchší a generalizovať lepšie na nové dáta, ale môže trpieť underfittingom.

Hyperparameter epsilon (ε) v epsilon-insensitive loss funkcii:

- Funkcia: Epsilon určuje šírku pásma okolo predikovaných hodnôt, v ktorom sa chyby neberú do úvahy. Toto pásmo sa nazýva epsilon-tube.
- Vplyv:
    - Vysoké hodnoty epsilon**:** Väčšie pásmo, kde sa chyby ignorujú. Model môže byť menej presný, pretože viacej chýb sa nezohľadňuje.
    - Nízke hodnoty epsilon**:** Menšie pásmo, čo znamená, že model sa snaží presne predpovedať hodnoty s menšími chybami. Môže to viesť k lepšiemu prispôsobeniu sa tréningovým dátam, ale tiež k vyššiemu riziku overfittingu.

Hyperparameter gamma (γ) v RBF a Poly kerneloch:

- Funkcia: Gamma určuje, ako ďaleko dosahuje vplyv jednotlivých tréningových príkladov. Ovláda polomer rozhodovacieho regiónu.
- Vplyv:
    - Vysoké hodnoty gamma**:** Každý tréningový príklad má malý dosah, čo vedie k veľmi flexibilnému modelu, ktorý sa môže prispôsobiť šumu v dátach a viesť k overfittingu.

- Nízke hodnoty gamma: Každý tréningový príklad má väčší dosah, čo vedie k hladkému modelu, ktorý môže lepšie generalizovať, ale môže trpieť underfittingom, ak je gamma príliš nízke.

Obr. 8: Trénovanie modelov

```
results = {}
predictions = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    results[name] = (mse, r2)
    predictions[name] = y_pred
    print(f"{name} - MSE: {mse}, R^2: {r2}")
```

Zdroj: Vlastné spracovanie

Blok kódu na obrázku 8 trénuje rôzne modely, predikuje hodnoty na testovacej množine, vypočíta metriky výkonu (*MSE a* R^2) a ukladá výsledky pre každý model. Najprv sú inicializované prázdne slovníky `results` a `predictions`. Potom sa iteruje cez všetky modely definované v slovníku models. Pre každý model sa vykoná tréning na tréningovej množine `X_train` a `y_train` pomocou metódy fit. Následne model predikuje hodnoty na testovacej množine `X test` pomocou metódy predict. Vypočítajú sa metriky výkonu: *MSE* (Mean Squared Error), ktorý meria priemerný štvorec rozdielov medzi skutočnými a predikovanými hodnotami, a R^2 (R-squared), ktorý meria, aká časť variability závislej premennej je vysvetlená nezávislými premennými. Tieto hodnoty sú uložené v slovníku results pod názvom modelu. Predikované hodnoty `y pred` sú uložené v slovníku predictions pod názvom modelu. Nakoniec sa výsledky pre každý model vytlačia vo formáte, ktorý zobrazuje názov modelu, hodnoty MSE a R^2.

Obr. 9: Výsledky MSE a koeficientov determinacie modelov

```
Linear Regression - MSE: 18487911.523133304, R^2: 0.8733561613662507
SVR (linear kernel) - MSE: 28873875.77526806, R^2: 0.8022113822949408
SVR (poly kernel) - MSE: 4948200.620824575, R^2: 0.9661043855512286
SVR (rbf kernel) - MSE: 3018203.824105105, R^2: 0.9793250353029085
SVR (sigmoid kernel) - MSE: 4334971259.047177, R^2: -28.69493876719227
```

Zdroj: Vlastné spracovanie

MSE je priemer štvorcov rozdielov medzi skutočnými a predikovanými hodnotami. Nižšie hodnoty *MSE* naznačujú lepší model, pretože rozdiely medzi skutočnými a predikovanými hodnotami sú menšie. Vyššie hodnoty *MSE* naznačujú horší model, pretože rozdiely medzi skutočnými a predikovanými hodnotami sú väčšie.R^2 meria, aká časť variability závislej premennej (skutočné hodnoty) je vysvetlená nezávislými premennými (predikované hodnoty). Hodnoty R^2 sa pohybujú od 0 do 1.

- Hodnota blízka 1: Model veľmi dobre vysvetľuje variabilitu dát. Vysoké hodnoty R^2 naznačujú, že model je presný.

- Hodnota blízka 0: Model nevysvetľuje variabilitu dát. Nízke hodnoty R^2 naznačujú, že model nie je presný.
- Negatívna hodnota R^2: Model je horší ako jednoduchý priemer skutočných hodnôt. To sa stáva, keď je model veľmi nepresný.

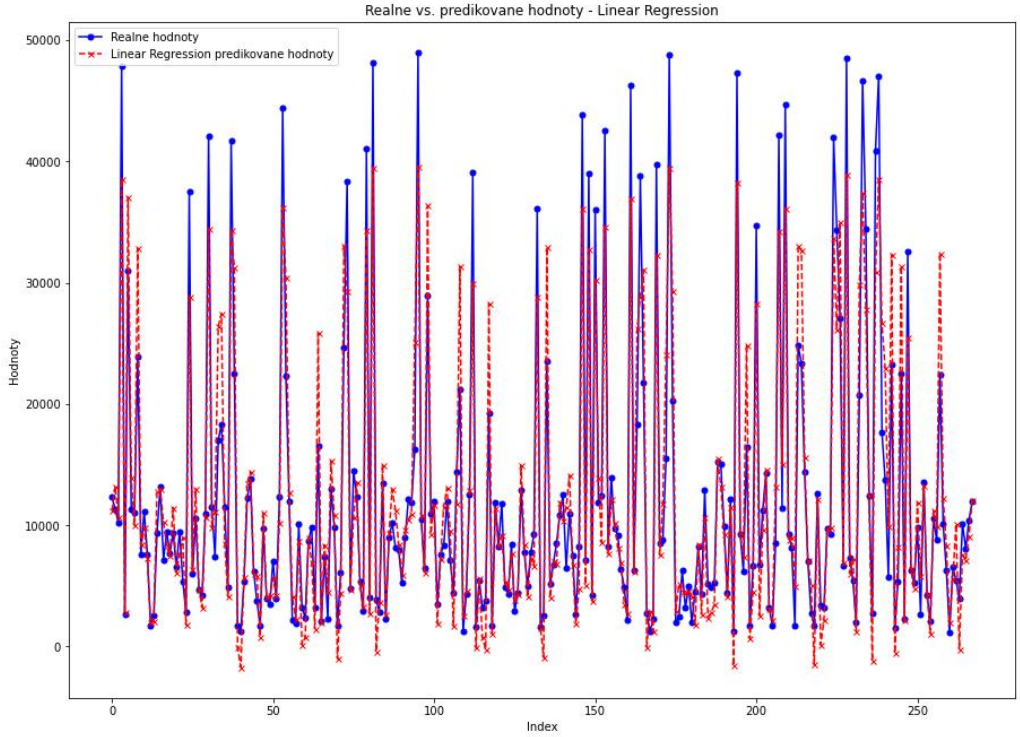
Na základe výsledkov zobrazených na obrázku 9, SVR s RBF kernelom dosahuje najlepší výkon, pretože má najnižšie MSE a najvyššie R^2, čo naznačuje, že najlepšie zachytáva zložité vzory v dátach. SVR s polynomiálnym kernelom tiež dosahuje dobrý výkon, ale nie taký dobrý ako RBF kernel. Lineárna regresia a SVR s lineárnym kernelom majú slušný výkon, ale sú horšie v porovnaní s RBF a poly kernelmi. SVR so sigmoid kernelom má veľmi zlý výkon a neodporúča sa pre tento dataset.

Obr. 10: Porovnanie skutočných a predikovaných hodnôt nákladov spolu s vysvetľujúcimi premennými pomocou modelu SVR (RBF kernel) pre náhodných 10 záznamov testovacej sady

age	bmi	children	smoker_yes	sex_male	region_northwest	region_southeast	region_southwest	Actual Charges	Predicted Charges (SVR rbf kernel)
23	23.180	2	False	False	True	False	False	3180.51010	3522.806766
53	22.610	3	True	False	False	False	False	24873.38490	22839.468760
38	40.150	0	False	False	False	True	False	5400.98050	5207.478675
26	32.900	2	True	True	False	False	True	36085.21900	31217.831135
31	25.900	3	True	True	False	False	True	19199.94400	19647.316551
31	25.800	2	False	False	False	False	True	4934.70500	4551.568099
24	32.700	0	True	True	False	False	True	34472.84100	32215.236296
37	30.875	3	False	True	True	False	False	6796.86325	6434.605838
46	35.530	0	True	False	False	False	False	42111.66470	42140.144084
26	35.420	0	False	True	False	True	False	2322.62180	2578.579903

Zdroj: Vlastné spracovanie

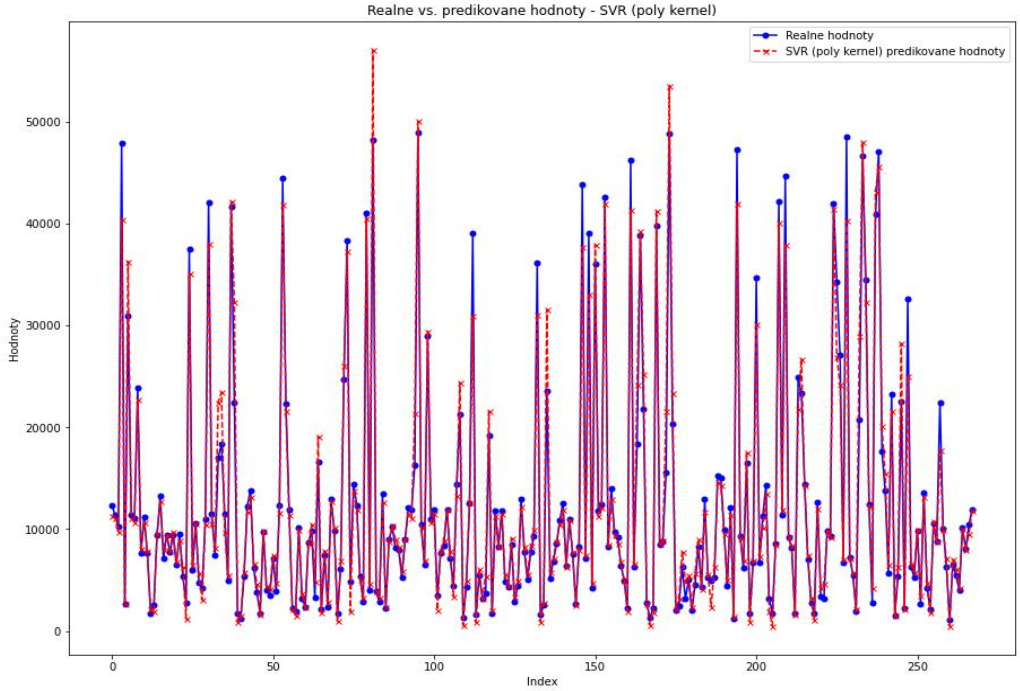
Obr. 11: Reálne vs. Predikované hodnoty(Lineárna regresia)



Zdroj: Vlastné spracovanie

Graf na obrázku 11 zobrazuje porovnanie reálnych a predikovaných hodnôt pomocou lineárnej regresie s koeficientom determinácie $R^2 = 0.873$.

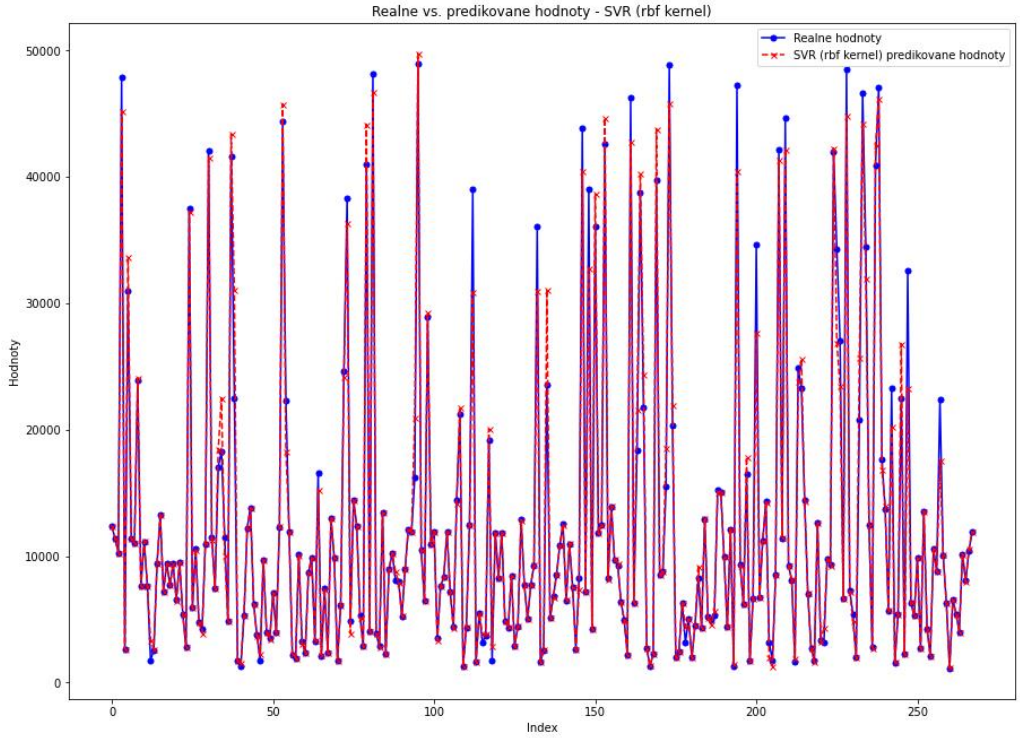
Obr. 12: Reálne vs. Predikované hodnoty(SVR s poly kernelom)



Zdroj: Vlastné spracovanie

Graf na obrázku 12 zobrazuje porovnanie reálnych a predikovaných hodnôt pomocou modelu SVR(kernel=poly) s koeficientom determinácie $R^2 = 0.966$.

Obr. 13: Reálne vs. Predikované hodnoty(SVR s RBF kernelom)



Zdroj: Vlastné spracovanie

Graf na obrázku 13 zobrazuje porovnanie reálnych a predikovaných hodnôt pomocou modelu SVR(kernel=RBF) s koeficientom determinácie $R^2 = 0.979$.

6. Záver

Podľa analýzy hodnôt MSE (Mean Squared Error) a koeficientu determinácie R^2 sa model Support Vector Regression (SVR) ukázal ako efektívnejší oproti lineárnej regresii pri odhade poistného. SVR modely s rôznymi kernelmi, najmä s RBF kernelom, poskytovali presné predikcie a efektívne vysvetľovali variabilitu v údajoch. Na druhej strane, lineárna regresia vykázala nižšiu presnosť a efektivitu v porovnaní s SVR modelmi. Výnimkou bol SVR so sigmoid kernelom, ktorý vykazoval najhoršie výsledky pravdepodobne z dôvodu, že na rozdiel od RBF kernelu, ktorý je veľmi efektívny pri zachytávaní nelineárnych vzťahov medzi dátovými bodmi, sigmoidálny kernel nemusí byť dostatočne flexibilný na zachytenie zložitých vzťahov v dátach.

Dôležité je pripomenúť, že účinnosť modelu SVR závisí od dôkladného nastavenia hyperparametrov, ako sú parametre C, epsilon a gamma, a vyžaduje presné škálovanie vstupných dát. Tieto parametre majú kritický vplyv na výkon modelu a vyžadujú starostlivú optimalizáciu. V prípade datasetu s poistnými nákladmi, SVR s RBF kernelom poskytol najpresnejšie predikcie s hodnotou R^2 0.9793, čo naznačuje, že bol schopný najlepšie vysvetliť variabilitu v údajoch. V konečnom dôsledku, aj keď je SVR časovo náročnejší na nastavenie a vyžaduje intenzívnejšie zdroje na optimalizáciu, jeho schopnosť poskytovať mimoriadne presné predikcie ho robí cenným vo vysoko špecializovaných aplikáciách, kde je kritická maximálna prediktívna presnosť a sú dostupné zdroje na jeho dôkladnú kalibráciu a údržbu.

Tento príspevok vznikol v rámci výskumného projektu VEGA 1/0431/22, *Implementácia inovatívnych prístupov modelovania rizík v procese ich riadenia v interných modeloch poisťovní v kontexte s požiadavkami direktívy Solvency II.*

Literatúra

1. Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.
2. Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing-Letters and Reviews*, *11*(10), 203-224.
3. Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), Article 27. https://doi.org/10.1145/1961189.1961199
4. Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, 36(3), 1171-1220. https://doi.org/10.1214/009053607000000677
5. Lins, I. D. et al, (2013). Sea Surface Temperature prediction via Support Vector Machines combined with Particle Swarm Optimization. *Expert Systems with Applications*, 40(5), 1766-1779.
6. Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge University Press.
7. Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199-222. https://doi.org/10.1023/B:STCO.0000035301.49549.8
8. Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc. https://doi.org/10.1007/978-1-4757-2440-0
9. Zhang, F., & O'Donnell, L. J. (2020). Support vector regression. *In Machine Learning* (pp. 123–140). Elsevier. https://doi.org/10.1016/B978-0-12-815739-8.00007-9
10. Zhu, J., & Hastie, T. (2005). Kernel Logistic Regression and the Import Vector Machine. *Journal of Computational and Graphical Statistics*, 14(1), 185–205. https://doi.org/10.1198/106186005X25619