Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-182905-105345

Bc. Monika Zjavková

Predikčný model na stanovenie dermatotoxicity

Diplomová práca

Vedúci práce: Ing. Marta Šoltésová Prnová PhD.

Máj 2025

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-182905-105345

Bc. Monika Zjavková

Predikčný model na stanovenie dermatotoxicity

Diplomová práca

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Informatika

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU

Vedúci práce: Ing. Marta Šoltésová Prnová PhD.

Máj 2025

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Inteligentné softvérové systémy

Autor: Bc. Monika Zjavková
Diplomová práca: Predikčný model na stanovenie dermatotoxicity
Vedúci diplomového projektu: Ing. Marta Šoltésová Prnová PhD.
Máj 2025

Diplomová práca sa zaoberá vývojom predikčného modelu na hodnotenie dermatotoxicity chemických látok, čo je dôležitá časť ochrany verejného zdravia. Dermatotoxicita predstavuje schopnosť chemických látok spôsobiť podráždenie alebo poškodenie pokožky, pričom jej tradičné hodnotenie zahŕňa nákladné a eticky problematické testy na zvieratách. S cieľom nahradiť tieto metódy sa v práci využíva prístup QSAR (kvantitatívny vzťah medzi štruktúrou a aktivitou), ktorý umožňuje predpovedať biologické vlastnosti látok na základe ich chemickej štruktúry. Súčasťou práce je analýza chemických deskriptorov, výber najrelevantnejších vlastností pomocou rôznych metód, a aplikácia algoritmov strojového učenia vrátane Decision tree, Random Forest, KNN, XGBoost a SVM.

Annotation

Slovak University of Technology Bratislava
Faculty of Informatics and Information Technologies
Degree Course: Inteligentné softvérové systémy

Author: Bc. Monika Zjavková
Master's Thesis: Predictive model for the determination of dermatotoxicity
Supervisor: Ing. Marta Šoltésová Prnová PhD.
Máj 2025

This master's thesis addresses the development of a predictive model for assessing the dermatotoxicity of chemical substances, a critical aspect of public health and safety. Dermatotoxicity refers to the ability of chemical compounds to cause skin irritation or damage, traditionally assessed through costly and ethically controversial animal testing. To replace these methods, the thesis applies the QSAR (Quantitative Structure-Activity Relationship) approach, which makes the prediction of biological properties based on chemical structure. The work involves analyzing chemical descriptors, selecting the most relevant features using techniques such as filter methods and wrapper approaches, and employing machine learning algorithms including Decision Tree, Random Forest, KNN, XGBoost, and SVM. Model validation and result interpretation were carried out using metrics like accuracy, recall, and F1 score.

Slovenská technická univerzita v Bratislave
Ústav informatiky, informačných systémov a softvérového inžinierstva

Fakulta informatiky a informačných technológií
Akademický rok: 2024/2025
Evidenčné číslo: FIIT-182905-105345

ZADANIE DIPLOMOVEJ PRÁCE

Študentka: **Bc. Monika Zjavková**
ID študenta: 105345
Študijný program: inteligentné softvérové systémy
Študijný odbor: informatika
Vedúca práce: Ing. Marta Šoltésová Prnová, PhD.
Vedúci pracoviska: doc. Ing. Ján Lang, PhD.

Názov práce: **Predikčný model na stanovenie dermatotoxicity**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Dermatotoxicita sa zaoberá štúdiom reakcií kože na chemické látky a hrá kľúčovú úlohu pri hodnotení bezpečnosti kozmetických výrobkov a chemických látok. Podľa OECD je možné testovať látky aj prostredníctvom prediktívnych modelov, ktorých cieľom je postupne nahradiť štandardné testy na zvieratách a bunkových kultúrach. Tieto modely predstavujú rýchlejšie a ekologickejšie alternatívy k tradičným in vivo testom, čo prispieva k etickejšiemu a efektívnejšiemu hodnoteniu dermatotoxicity. Analyzujte existujúce prístupy zaoberajúce sa predikciou iritácie kože. Porovnajte viaceré prístupy k analýze deskriptorov a zvoľte vhodnú metódu na ich výber. S využitiím dátovej vedy a strojového učenia navrnite predikčný model pre vyhodnotenie vplyvu látky na pokožku pomocou analýzy vzťahu štruktúry a účinku látky. Vyberte vhodnú metódu strojového učenia, vysvetlite rozhodovací algoritmus modelu a zabezpečte jeho interpretovateľnosť. Overte vytvorený model a vyhodnoťte jeho úspešnosť na vzorke existujúcich dát.

Termín odovzdania diplomovej práce: 11. 05. 2025
Dátum schválenia zadania diplomovej práce: 15. 04. 2025
Zadanie diplomovej práce schválil: prof. Ing. Vanda Benešová, PhD. – garantka študijného programu

Čestne vyhlasujem, že som túto prácu vypracoval(a) samostatne, na základe konzultácií a s použitím uvedenej literatúry.

Bratislava 22.5.2023 .....

Bc. Monika Zjavková

Poďakovanie

Chcela by som poďakovať všetkým osobám, ktoré mi svojou podporou a pomocou prispeli k úspešnému dokončeniu tejto diplomovej práce. Predovšetkým by som chcela poďakovať vedúcej tejto diplomovej práce Ing. Marte Šoltésovej Prnovej, PhD. za jej odborné vedenie, cenné rady, trpezlivosť a ochotu počas celého obdobia riešenia tejto práce.

Obsah

1 Analýza problému **3**

1.1 Kvantitatívna analýza vzťahu štruktúry a účinku 3

1.2 Chemické deskriptory 5

1.2.1 Výber deskriptorov 8

1.2.1.1 Filtračné Metódy 8

1.2.1.2 Wrapper metódy 8

1.2.1.3 Embedded metody 9

1.2.1.4 SHAP 10

1.3 Modely strojového učenia 11

1.3.1 Rozhodovací strom 12

1.3.2 Náhodný les 13

1.3.3 XGBOOST 13

1.3.4 K-susedia 14

1.3.5 SVM 16

1.3.6 Metriky vyhodnocovania 17

1.4 Vysvetliteľnosť 18

1.5 Existujúce riešenia 19

1.5.1 Ensemble learning 19

1.5.2 Podráždenia pokožky hlbokých eutektických rozpúšťadiel . . 20

1.5.3 QSAR Toolbox 21

2 Návrh riešenia **23**

2.1 Špecifikácia požiadaviek 23

2.1.1 Funkcionálne požiadavky 24

2.1.2 Nefunkcionálne požiadavky 24

2.2 Prípady použitia 25

2.2.1 Vloženie látky na zistenie predikcie 26

2.2.2 Predspracovanie a výpočet deskriptorov 27

2.2.3 Predikcia dermatotoxicity 27

2.2.4 Vizualizácia a vysvetliteľnosť 27

2.3 Workflow 28

2.4 Architektúra riešenia 31

2.4.1 Frontend (Užívateľské rozhranie) 31

2.4.2 Backend (Spracovanie a predikcia) 31

2.5 Výber metód strojového učenia 32

2.5.1 Decision tree (Rozhodovací strom) 32

2.5.2 Random Forest (Náhodný les) 33

2.5.3 XGBoost (Extreme Gradient Boosting) 33

2.5.4 Support Vector Machine (SVM) 34

2.5.5 KNN 34

3 Implementácia riešenia **35**

3.1 Dataset 35

3.2 Výpočet deskriptorov 36

3.3 Predspracovanie dát 37

3.4 Implementácia Algoritmov 37

3.5 Výber najvhodnejších deskriptorov 38

3.5.1 Výber na základe korelácie a rozptylu 38

3.5.2 Výber deskriptorov pomocou Lasso regresie 39

3.5.3 Výber podľa SHAP hodnoty 40
3.5.3.1 Výber deskriptorov v Rozhodovacom strome 41
3.5.3.2 Výber deskriptorov v SVM 42
3.5.3.3 Výber deskriptorov v Náhodnom lese 43
3.5.3.4 Výber deskriptorov v XGBoost 44
3.5.3.5 Výber deskriptorov v KNN 45
3.5.4 Zhodnotenie výberu deskriptorov 46
3.6 Optimalizácia 48
3.6.1 XGBoost (Extreme Gradient Boosting) 49
3.6.2 Rozhodovací strom (Decision Tree) 50
3.6.3 Náhodný les (Random forest) 52
3.6.4 SVM (Support Vector Machine) 53
3.6.5 K-Nearest Neighbors (KNN) 54
3.6.6 Výsledky optimalizácie hyperparametrov 56
3.7 Optimalizácia datasetu 58
3.7.1 Zníženie počtu vzoriek pomocou klastrovania 58
3.7.2 Výsledky po optimalizácii datasetu 58
3.8 Spojenie modelov 59
3.8.1 Spojenie pomocou metamodelu 59
3.8.2 Hlasovanie 61
3.9 Webové rozhranie aplikácie 62
3.9.1 Backendová časť aplikácie 63
3.9.2 Frontendová časť aplikácie 64
3.9.3 Interakcia medzi komponentmi 66
3.10 Možné vylepšenia a budúci vývoj 68

A Inštalačná príručka **A-3**

B Používateľská príručka **A-5**

B.1 Predikcia podráždenia pokožky A-5

C Technická dokumentácia **A-7**

C.1 Fáza vývoja modelov (offline spracovanie) A-7

C.2 Webová aplikácia (Django) A-8

C.3 Použité knižnice A-9

D Časový plán práce **A-11**

D.1 Letný semester 2023/2024 . . . . . A-11

D.2 Zimný semester 2024/2025 . . . . . A-12

D.3 Letný semester 2024/2025 . . . . . A-13

D.4 Zhodnotenie časového harmonogramu A-13

E Opis elektronického média **A-17**

Úvod

Denne prichádzame do kontaktu s chemickými látkami z priemyselných, kozmetických či farmaceutických produktov. Koža funguje ako rozhodujúca bariéra voči vonkajšiemu prostrediu, vrátane týchto chemických látok. Avšak interakcia medzi pokožkou a chemickými zlúčeninami môže viesť k dermatotoxickým reakciám a môže nastať iritácia alebo poškodenie kože.

Pochopenie a predpovedanie dermatotoxicity chemických látok je preto dôležité pre ochranu verejného zdravia. Bežné metódy hodnotenia dermatotoxicity zahŕňajú testovanie na zvieratách, na bunkových alebo tkanivových kultúrach. Tieto prístupy sú však nielen eticky problematické vzhľadom na používanie zvierat, ale sú aj finančne a časovo náročné.

Pokroky v oblasti prediktívnej toxikológie prinášajú alternatívne metódy hodnotenia dermatotoxicity látok. Modely kvantitatívneho vzťahu medzi štruktúrou a aktivitou (QSAR) ponúkajú nástroj na pochopenie vzťahu medzi chemickou štruktúrou látok a ich biologickými účinkami.

Cieľom tejto práce je navrhnúť QSAR model na predikciu dermatotoxicity chemických látok na základe molekulárnych deskriptorov. Tieto deskriptory reprezentujú štruktúrne a fyzikálno-chemické vlastnosti zlúčenín, z ktorých sa vyberajú tie najvýznamnejšie pre tvorbu modelu. Výber je následne využitý v kombiná-

cii s metódami strojového učenia na predikciu potenciálneho dermatotoxického účinku.

Kapitola 1

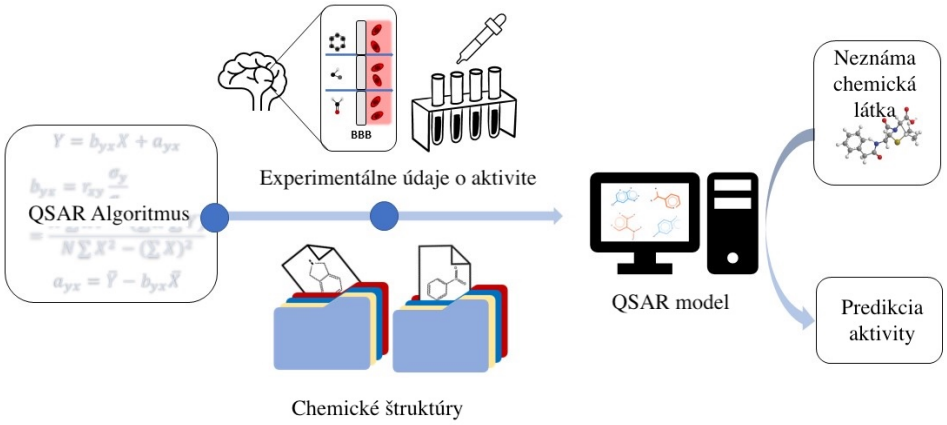
Analýza problému

Dermatotoxicita je reakcia pokožky na chemické látky a je dôležitou súčasťou v toxikológii. Porozumenie reakcie kože na vonkajšie prostredie je dôležité na vyhodnotenie potenciálnych rizík pri vystavení kože rôznym chemikáliám.

Klasické metódy hodnotenia dermatotoxicity zahŕňajú *in vivo* testy na zvieratách, na bunkových alebo tkanivových kultúrach, ale regulačné agentúry začínajú podporovať vývoj a používanie alternatívnych metód, ktoré sú nielen efektívnejšie, ale aj etické a lacnejšie. Toto viedlo k zavádzaniu výpočtových metód, najmä modelov kvantitatívnej analýzy vzťahu štruktúry a účinku, ktoré sa ukázali ako vhodné nástroje na predpovedanie dermatotoxicity.

1.1 Kvantitatívna analýza vzťahu štruktúry a účinku

Kvantitatívna analýza vzťahu štruktúry a účinku z angličtiny Quantitative Structure-Activity Relationship (QSAR) je využitie metód dátovej vedy a štatistík na vývoj modelov, ktoré predpovedajú biologické aktivity alebo vlastnosti zlúčenín na základe ich štruktúr [41].



Obr. 1.1: Proces QSAR [42].

Modely QSAR majú využitie pri hodnotení potenciálnych vplyvov chemikálií, materiálov a nanomateriálov na ľudské zdravie, tiež sa používajú ako nástroj na predpovedanie alebo navrhovanie nových chemikálií so špecifickými požadovanými vlastnosťami [8].

Proces kvantitatívneho vzťahu medzi štruktúrou a aktivitou (QSAR) zahŕňa zbieranie a prípravu údajov o chemických štruktúrach a príslušných aktivitách, z ktorých sa následne vypočítavajú molekulárne deskriptory. Nepoužívajú sa všetky deskriptory, ale vyberie sa iba niekoľko, ktoré majú najväčšiu výpovednú hodnotu. Potom nasleduje rozdelenie údajov na trénovacie, validačné a testovacie sady, vývoj prediktívneho modelu, validácia a optimalizácia výkonu modelu [41].

Modely QSAR možno rozdeliť na klasifikačné a regresné. Klasifikačné modely sa využívajú na triedenie zlúčenín do rôznych kategórií. Regresné modely sú často využívané na predpovedanie účinnosti.

Pri tvorbe QSAR modelov je dôležité zvoliť správny typ modelu podľa cieľového problému. Napríklad klasifikačné modely sú vhodné na binárne rozhodnutia (toxická vs. netoxická), zatiaľ čo regresné modely umožňujú presné kvantitatívne predikcie. Voľba medzi nimi závisí od dostupných dát a ich konkrétnemu využitiu [4].

Používanie QSAR modelov je regulované medzinárodnými usmerneniami vypracovanými Organizáciou pre hospodársku spoluprácu a rozvoj (OECD). Tieto usmernenia definujú kritériá, ktoré musia modely QSAR spĺňať, aby boli akceptované pre regulačné účely. Cieľom je zabezpečiť ich spoľahlivosť a reprodukovateľnosť.

Modely QSAR musia spĺňať niekoľko kritérií. Jedným z týchto kritérií je jasne definovaný koncový bod, napríklad predikcia iritácie pokožky alebo akútnej toxicity, ktorý musí byť merateľný a vedecky podložený. Rovnako dôležitý je jednoznačný algoritmus, ktorý je reprodukovateľný a transparentný, čo znamená, že jeho parametre, ako počet stromov alebo maximálna hĺbka v prípade Náhodného lesu, musia byť presne špecifikované, aby sa model dal opakovane vytvoriť a testovať. QSAR model musí taktiež špecifikovať svoju oblasť použiteľnosti, teda aké typy chemických zlúčenín alebo biologických vlastností dokáže presne predpovedať.

Okrem toho je nevyhnutné hodnotiť presnosť a predvídateľnosť modelu pomocou metrík, ako sú *accuracy* (presnosť), *precision* (precíznosť), *recall* (citlivosť) a F1-Skóre.[28].

1.2 Chemické deskriptory

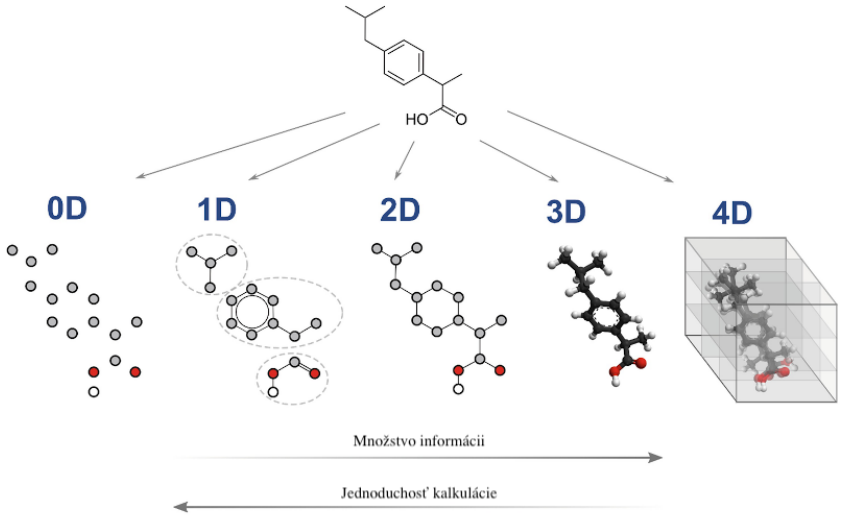
V oblasti chemickej informatiky sa molekuly reprezentujú pomocou matematických deskriptorov, ktoré slúžia na zakódovanie molekulárnej štruktúry a ich vlastností do vhodného formátu na spracovanie. Na definovanie vzťahov medzi týmito

deskriptormi a cieľovými vlastnosťami sa následne využívajú štatistické metódy alebo metódy strojového učenia. Informácie kódované deskriptormi vo všeobecnosti závisia od druhu molekulovej reprezentácie a definovaného algoritmu na jej výpočet. Niektoré z nich zahŕňajú geometrické, konštitučné a fyzikálno-chemické deskriptory [27, 10].

Konštitučné deskriptory sú jednoduché a často používané. Poskytujú pohľad na molekulárne zloženie zlúčeniny bez ohľadu na informácie o jej topológii [36]. Príklady konštitučných deskriptorov zahŕňajú základné charakteristiky, ako je počet atómov, počet väzieb, typ atómu, počet kruhov a molekulová hmotnosť.

Okrem konštitučných deskriptorov existujú aj topologické indexy, ktoré hovoria o konektivite atómov v molekule, čo pomáha pri hodnotení jej štrukturálnych vlastností. Geometrické deskriptory zahŕňajú parametre súvisiace s tvarom a priestorovým usporiadaním molekuly, ktoré podávajú informácie o jej trojrozmernej štruktúre. Fyzikálnochemické deskriptory zahŕňajú vlastnosti, ako je rozpustnosť, kyslosť a elektronegativita.

Ďalší typ delenia deskriptorov je podľa dimenzií. Tieto typy sú od 0D do 4D, predstavujú rôzne dimenzie informácií o chemických zlúčeninách. Postupujú od 0D zahŕňajúce základné fyzikálne a chemické vlastnosti, k singulárnym vlastnostiam v 1D, medziatómovým vzťahom v 2D, trojrozmerným štrukturálnym aspektom v 3D a dynamickým alebo časovo závislým charakteristikám v 4D [10].



Obr. 1.2: Chemické deskriptory [6].

Vzhľadom na zvýšenie výpočtovej sily hardvéru, softvéru a znižovanie nákladov na výpočtovú techniku a zber rôznych molekulárnych deskriptorov je dnes mnoho modelovania QSAR závislé od správnej analýzy a výberu vypočítaných deskriptorov ako nezávislých premenných pre tvorbu modelu QSAR [36].

Optimálne deskriptory na zostavenie modelov QSAR sú založené na zápise SMILES (Simplified Molecular Input Line Entry System) [45]. Predstavujú textový zápis molekuly, kde každý atóm a väzba sú reprezentované jedinečným znakom alebo skupinou znakov. Pri zostavovaní QSAR modelov umožňujú tieto SMILES deskriptory efektívne zachytiť rôznorodosť chemických štruktúr a ich vzájomné vzťahy, čo vedie k presnejším predpovediam biologickej aktivity [33].

Optimálne deskriptory pre tvorbu QSAR modelov sú odvodené zo zápisu SMILES (Simplified Molecular Input Line Entry System) [45]. Predstavujú textový zápis molekuly, kde každý atóm a väzba sú reprezentované jedinečným znakom alebo skupinou znakov. [33].

1.2.1 Výber deskriptorov

Hoci je technicky možné vytvoriť modely využívajúce všetky dostupné molekulárne deskriptory, z hľadiska presnosti predikcie je efektívnejšie zamerať sa len na tie najrelevantnejšie. Zo SMILES zápisov možno získať aj tisíc rôznych deskriptorov, čo môže viesť k nadmernej komplexnosti modelu. [14, 17].

Taktiež je model QSAR, ktorý sa má zostaviť, často jednoduchší a potenciálne rýchlejší, keď sa použije menej vstupných deskriptorov. Je to preto, že mnohé algoritmy strojového učenia majú výpočtovú zložitosť, ktorá rastie nelineárne s počtom vstupných premenných [10].

Pri výbere príznakov (*features*) v modeli existuje viacero prístupov ako filtračné metódy, wrapper, embeded a iné.

1.2.1.1 Filtračné Metódy

Filtračné metódy posudzujú deskriptory bez ohľadu na učiaci sa algoritmus alebo jeho výsledky. Hodnotia ich na základe kritérií, ako je napríklad korelácia, rozptyl a iné. Po vyhodnotení sa vyberú tie najdôležitejšie.

Patria tu metódy ako Chi-square test, ANOVA F-value alebo Fisher Score. Ich výhodou je, že sú rýchle a vhodné pre predbežnú analýzu veľkých množín údajov. Neposkytujú však informácie o interakciách medzi vlastnosťami a ich kombinovanom vplyve na model [19].

1.2.1.2 Wrapper metódy

Ďalším spôsobom sú Wrapper metódy, ktoré naopak vyberajú deskriptory na základe výkonu modelu, ktorý sa na nich trénuje. Používajú iteratívny prístup, kde sa pridávajú alebo odoberajú vlastnosti a sleduje sa zmena v presnosti modelu [17, 11].

Príklady wrapper metód sú napríklad popredný výber (*Forward selection*), spätná eliminácia (*Backward Elimination*) alebo postupná regresia, ktorá využíva popredný výber aj spätnú elimináciu.

Pri metóde popredného výberu sa model buduje postupne, na začiatku neobsahuje žiadne deskriptory a tie sa pridávajú jeden po druhom. V každom kroku sa vyberie ten deskriptor, ktorý najviac zlepší výkonnosť modelu, napríklad zvýši presnosť klasifikácie alebo zníži chybu predikcie. Tento proces pokračuje, kým pridávanie ďalších deskriptorov už neprináša významné zlepšenie.

Naopak, spätná eliminácia začína s modelom, ktorý obsahuje všetky dostupné deskriptory. Tie sa potom postupne odstraňujú — v každom kroku sa odstráni ten, ktorý najmenej prispieva k výkonu modelu, často na základe štatistických kritérií ako je súčet štvorcov chýb. Proces sa ukončí, keď zostanú len dôležité deskriptory, alebo keď ďalšie odstránenie vedie k zhoršeniu výkonnosti modelu [36, 14].

Na rozdiel od filtračných metód už zohľadňujú interakcie medzi atribútmi(features) a optimalizujú výber pre konkrétny model. Sú však časovo náročné a môžu byť zložité na výpočet, hlavne pri veľkých dátových množinách [13, 14].

1.2.1.3 Embedded metody

Embedded metódy vykonávajú výber deskriptorov priamo počas tréningu modelu. Tým sa stávajú súčasťou algoritmu učenia a umožňujú, aby výber deskriptorov prebiehal súčasne s modelovaním, čo vedie k ich efektívnejšiemu a presnejšiemu výberu.

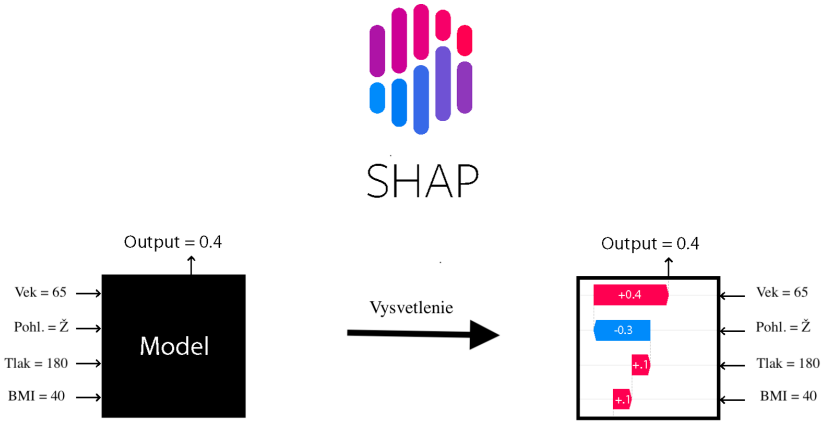
Medzi tieto metódy patria regularizované techniky, ako sú Lasso a Ridge Regression. Tieto metódy sú efektívne, pretože sú optimalizované tak, aby zlepšili výkon konkrétneho modelu, s ktorým sú spojené. Ich účinnosť však môže klesnúť, ak sa vybrané deskriptory použijú s iným modelom, pretože výber bol špecificky prispô-

sobený pôvodnému algoritmu [19].

1.2.1.4 SHAP

SHAP hodnoty, známe tiež ako Shapley Additive Explanations, sú metódou interpretácie výstupov modelov strojového učenia, ktorá vychádza z kooperatívnej teórie hier. Konkrétne, Shapley hodnoty sú pôvodne navrhnuté na rozdelenie celkového prínosu medzi hráčov v kooperatívnej hre tak, aby bola zachovaná spravodlivosť. V kontexte strojového učenia sa tieto hodnoty využívajú na hodnotenie príspevku jednotlivých vstupných príznakov k predikcii konkrétneho modelu.

Súčet príspevkov alebo hodnôt SHAP každého prvku sa rovná konečnej predikcii. V tomto prípade hodnota SHAP nie je rozdielom medzi predikciou s funkciou a bez nej, ale je príspevkom funkcie k rozdielu medzi skutočnou predpoveďou a strednou predikciou. Výsledný graf ukazuje negatívne (modrou) a kladnné (červenou) hodnoty SHAP, ktoré znižujú a zvyšujú predpoveď modelu [23].



Obr. 1.3: SHAP hodnoty modelu [37].

SHAP hodnoty ukazujú akú veľkú úlohu zohráva jednotlivý atribút v rozhodovacom procese modelu. Na rozdiel od tradičných metód, ktoré často zvažujú len

priame účinky, SHAP hodnoty zohľadňujú aj interakcie medzi jednotlivými premennými, čím poskytujú presnejší pohľad na to, ako model pracuje. Toto je obzvlášť dôležité pri zložitých modeloch, ako sú napríklad neurónové siete alebo metódy súborového učenia (*ensemble learning*), kde môže byť ťažké pochopiť rozhodovacie procesy [29].

Pri modeloch strojového učenia sa SHAP hodnoty používajú na rôzne účely, vrátane vysvetľovania individuálnych predikcií, porovnávania významnosti premenných a diagnostiky modelu. Jednou z kľúčových aplikácií SHAP hodnôt je feature selection, teda výber relevantných vstupných príznakov. V QSAR modeloch pomocou SHAP hodnôt môžeme identifikovať, ktoré štruktúrne charakteristiky majú najväčší vplyv na biologickú aktivitu, čo nám umožňuje zlepšiť výkonnosť modelu a lepšie pochopiť vzťah medzi štruktúrou a aktivitou zlúčenín [23].

1.3 Modely strojového učenia

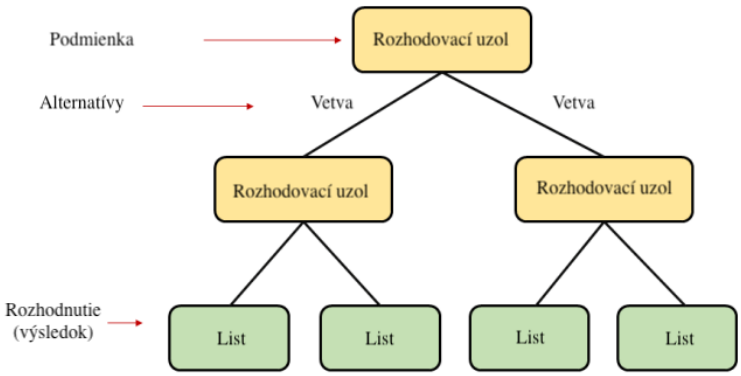
Existuje veľa algoritmov strojového učenia, ktoré sa úspešne aplikujú v QSAR modelovaní. Tieto algoritmy sa líšia vo svojej komplexnosti, princípoch fungovania a efektivite pri spracovaní dát. Medzi najvyužívanejšie patria XGBoost, K-Nearest Neighbors (k-NN), Random Forest, Decision Tree a Support Vector Machine (SVM). Každý z týchto algoritmov má svoje špecifické vlastnosti, výhody a obmedzenia, ktoré ich robia vhodnými pre rôzne typy úloh v oblasti QSAR.

Nasledujúce sekcie detailne opisujú každý z týchto algoritmov, ich základné princípy, spôsob fungovania a aplikáciu v QSAR modelovaní.

1.3.1 Rozhodovací strom

Rozhodovacie stromy patria medzi základné algoritmy strojového učenia, ktoré sa často využívajú pri klasifikácii. Tieto algoritmy fungujú na princípe postupného delenia dát na homogénne skupiny, pričom každý uzol v strome predstavuje rozhodovanie na základe určitej vlastnosti alebo atribútu, a každý list stromu reprezentuje konečný výsledok (triedu alebo predikciu).

Výhodou rozhodovacích stromov je ich interpretovateľnosť. Na jednotlivých vetvách stromu je možné vysvetliť, na základe akých kritérií bola vykonaná klasifikácia.



Obr. 1.4: Rozhodovací strom [5].

Jedným z hlavných problémov rozhodovacích stromov je ich sklon k pretrénovaniu, najmä ak sú stromy veľmi hlboké. Tento problém sa často rieši metódami ako napríklad prerezávanie stromov alebo využitím ensemble prístupov, akými sú náhodné lesy (Random Forest) [40].

1.3.2 Náhodný les

Random Forest (náhodný les) patrí do skupiny ensemble metód, čo znamená, že model kombinuje viacero rozhodovacích stromov a využíva kombináciu ich rozhodnutí na dosiahnutie konečnej predikcie [31].

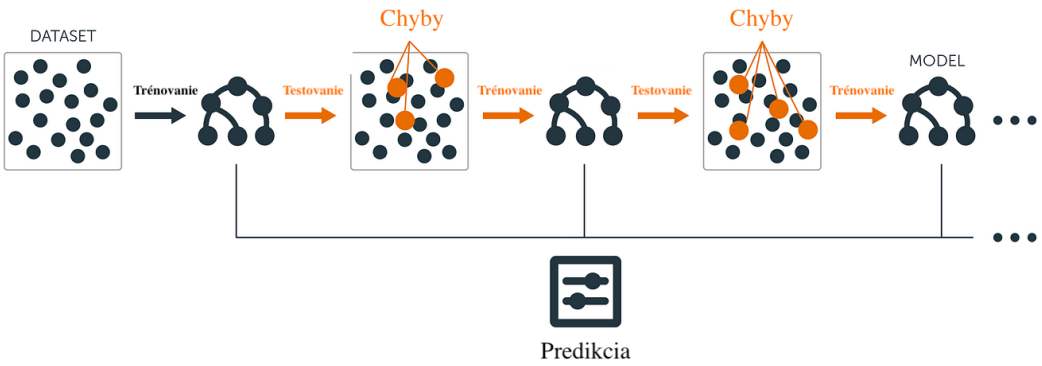
Fungovanie Náhodného lesu je založené na princípe rozhodovacích stromov. Pre každý strom sa použije náhodná podmnožina trénovacích dát a náhodný výber podmnožiny funkčných vlastností. Tento náhodný výber má za následok vytvorenie rôznych stromov, ktoré sú často menej náchylné na pretrénovanie ako jeden veľký strom [34].

Náhodný les ponúka niekoľko výhod, vrátane možnosti určiť dôležitosť rôznych funkčných vlastností, schopnosti zachovať vysokú presnosť aj pri veľkých datasetoch a odolnosti voči pretrénovaniu. Tieto vlastnosti robia z Náhodného lesu vhodný nástroj pre mnoho aplikácií v oblasti chemoinformatiky a QSAR analýz [39].

1.3.3 XGBOOST

XGBoost (eXtreme Gradient Boosting) je algoritmus strojového učenia, ktorý sa stal populárnym v mnohých oblastiach, vrátane QSAR modelovania. Je založený na technike ensemble learning, ktorá kombinuje viacero slabších modelov (v tomto prípade rozhodovacie stromy) a vytvára tak silný model so schopnosťou predikovať presnejšie výsledky [46].

Základným princípom XGBoostu je iteratívne zlepšovanie predikcie pomocou pridávania stromov do ensemble modelu. Každý nový strom je trénovaný tak, aby upravoval chyby predchádzajúcich stromov. Tento postup je realizovaný pomocou gradientovej optimalizácie, kde sa minimalizuje chyba predikcie.



Obr. 1.5: XGBoost [20].

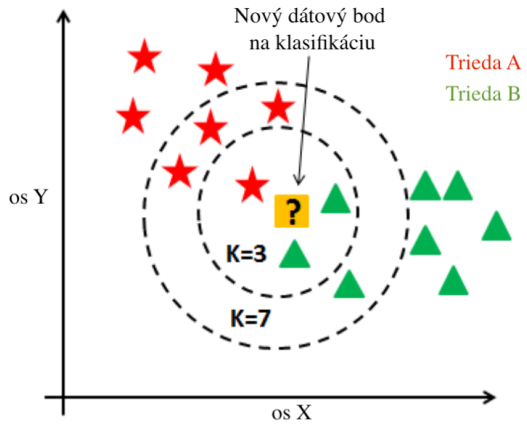
Model predikuje výslednú hodnotu ako váhovú kombináciu predikcií jednotlivých stromov, pričom každý strom prispieva k výslednej predikcii s určitou váhou. Táto váha sa určuje na základe toho, ako dobre strom predikuje chyby predchádzajúcich stromov [7].

XGBoost patrí medzi často používané algoritmy v QSAR modelovaní najmä vďaka schopnosti efektívne spracovávať veľké množstvá molekulárnych deskriptorov a generovať presné predikcie biologickej aktivity zlúčenín. Jeho silnou stránkou je schopnosť modelovať aj zložité nelineárne vzťahy. Tento výkon je podporený technikami regularizácie, ktoré sú súčasťou algoritmu a pomáhajú predchádzať preučeniu modelu, čím zároveň zlepšujú jeho schopnosť generalizovať na nové dáta [12].

1.3.4 K-susedia

K-Nearest Neighbors (k-NN) je jednoduchý algoritmus používaný v rôznych oblastiach analýzy dát, vrátane chemoinformatiky a QSAR. Jeho princíp spočíva v tom, že predikuje hodnoty pre nové dáta na základe ich podobnosti s dátami v trénovacej množine [26].

Pre daný bod v priestore funkčných vlastností (často nazývaným "dátový bod"), k-NN vyhľadá k najbližších susedov v trénovacej množine. Počet susedov (k) je definovaný používateľom. Potom sa použije metrika podobnosti (napríklad euklidovská vzdialenosť) na vyhodnotenie, akí susedia sú najviac podobní [47].



Obr. 1.6: KNN [2].

V QSAR sa k-NN používa často, keďže predpokladáme, že chemicky podobné zlúčeniny majú tendenciu mať podobné biologické vlastnosti. Samotný algoritmus nevyžaduje žiadne predpoklady o distribúcii dát, čo je výhodné pri analýze chemických dát, kde môžu byť vzorky zložité a nehomogénne.

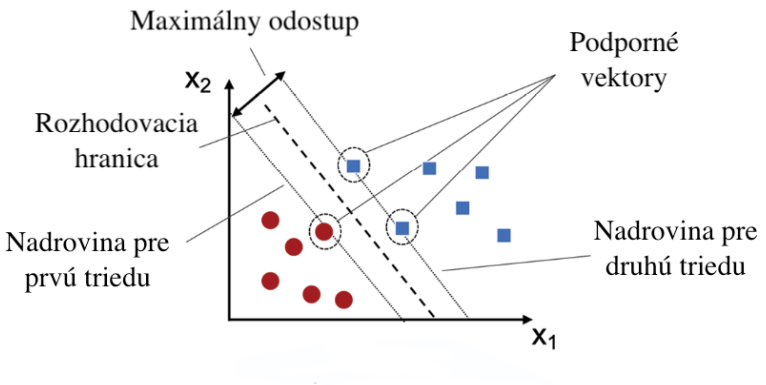
Avšak, výber správneho parametra k (počet susedov) je dôležitý, pretože ovplyvňuje presnosť predikcií a vyváženosť medzi biasom (skreslením) a varianciou modelu. Bias (skreslenie) je systematická chyba, ktorá vzniká, keď model nedostatočne reprezentuje vzory v dátach, čo vedie k trvalému odchýleniu predikcií od skutočných hodnôt. [30, 48].

V praxi môže byť implementácia algoritmu k-NN pomerne jednoduchá a efektívna, najmä pre malé a stredne veľké datasety. Avšak, pri veľkých datasetoch môže vyhľadávanie najbližších susedov byť časovo náročné a vyžadovať efektívne metódy

indexácie dát [48].

1.3.5 SVM

Support Vector Machine (SVM) je algoritmus strojového učenia, používaný na klasifikáciu aj regresiu. Jeho hlavným princípom je hľadať hranicu medzi triedami, ktorá maximalizuje odostup (margin) medzi najbližšími bodmi jednotlivých tried. Tento prístup umožňuje efektívne oddeliť dáta do rôznych kategórií s minimálnymi chybami.



Obr. 1.7: SVM [43].

Algoritmus funguje na princípe vyhľadávania optimálnej nadroviny (Hyperplane), ktorá čo najlepšie rozdeľuje dáta na základe ich triednej príslušnosti. Nadrovina je definovaná ako lineárna kombinácia podporných vektorov—bodov v priestore funkčných vlastností, ktoré sú najbližšie k hranici medzi triedami [18].

Pre nelineárne problémy môže byť SVM rozšírený pomocou kernelu. Tento prístup umožňuje mapovať vstupné dáta do viacdimenzionálneho priestoru, kde sa dajú modelovať zložité nelineárne vzťahy medzi dátami [32].

V QSAR analýzach sa SVM často používa pre svoju schopnosť efektívne pracovať s obmedzeným množstvom trénovacích dát. [25].

1.3.6 Metriky vyhodnocovania

Pri vyhodnocovaní výkonnosti binárnej klasifikácie je dôležité používať rôzne metriky, ktoré poskytujú celkový pohľad na presnosť modelu. Niektoré z najpoužívanejších metrík sú *accuracy* (presnosť), *precision* (precíznosť), *recall* (citlivosť) a F1-Skóre.

Presnosť (*accuracy*) vyjadruje pomer správne klasifikovaných príkladov ku všetkým príkladom v teste, zatiaľ čo *precision* meria percento pravdivo pozitívnych príkladov medzi všetkými pozitívnymi príkladmi. *Recall* predstavuje percento pravdivo pozitívnych príkladov, ktoré boli správne identifikované medzi všetkými skutočne pozitívnymi príkladmi. F1-Skóre je harmonický priemer medzi *accuracy* a *recall* a poskytuje vyváženú metriku pre hodnotenie výkonnosti modelu [44].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

Okrem základných metrík, ako sú presnosť, precíznosť či F1 skóre, sa pri hodnotení

klasifikačných modelov často využíva aj ROC krivka (Receiver Operating Characteristic Curve) a jej plocha – ROC AUC (Area Under the Curve). ROC krivka znázorňuje závislosť medzi mierou skutočne pozitívnych prípadov (true positive rate, tzn. citlivosť) a mierou falošne pozitívnych prípadov (false positive rate, tzn. 1 – špecificita) pri rôznych nastaveniach rozhodovacieho prahu [16].

Pri niektorých modeloch, ako je SVM (Support Vector Machine) a XGBoost, sa používa aj metrika Log Loss, ktorá vyjadruje chybu klasifikácie v kontexte pravdepodobností predikcií. Ďalšie metriky, ako je konfúzna matica, umožňujú podrobnejšie vyhodnotenie správnosti a chybovosti klasifikácie pre jednotlivé triedy [22, 44].

1.4 Vysvetliteľnosť

Vysvetliteľnosť v QSAR modeloch sa týka schopnosti pochopiť, interpretovať a vysvetliť rozhodnutia, ktoré tieto modely robia. Je to dôležitý aspekt v oblasti chemoinformatiky, aby bolo možné použiť výsledky modelov [15].

Na interpretáciu modelov QSAR bolo vyvinutých množstvo prístupov, ktoré možno rozdeliť do dvoch kategórií: tie, ktoré sú aplikovateľné na konkrétne modely strojového učenia, a tie, ktoré sú použiteľné na akékoľvek modely.

Interpretačné prístupy možno klasifikovať aj podľa úrovne interpretácie: interpretácia založená na vlastnostiach alebo štrukturálna interpretácia. V prístupoch založených na vlastnostiach sa počítajú príspevky alebo dôležitosť jednotlivých vlastností/deskriptorov. Tieto informácie už môžu byť užitočné, ak sú deskriptory samy osebe interpretovateľné [24].

SHAP vysvetľuje predikcie tým, že ukazuje, ako každá funkčná vlastnosť ovplyvňuje výslednú predikciu modelu. Výstupom metódy SHAP sú grafy, ktoré zobra-

zujú prínosy jednotlivých vlastností k predikciám. Tieto grafy môžu byť užitočné pri pochopení, ktoré vlastnosti majú najväčší vplyv na výsledné predikcie a ako sa mení predikcia v závislosti od zmien vstupných hodnôt [35].

Okrem SHAP existuje aj niekoľko ďalších metód na vysvetlenie predikcií modelov, vrátane LIME (Local Interpretable Model-agnostic Explanations), Partial Dependence Plots (PDP) a Accumulated Local Effects (ALE) Plots. Tieto metódy majú za cieľ poskytnúť užívateľom lepšie porozumenie ich modelov a dôveru v ich predikcie. Pomáhajú identifikovať dôležité funkčné vlastnosti, odhaľovať vzory v dátach a odhaľovať potenciálne slabé miesta v modeloch [1].

1.5 Existujúce riešenia

Vo výskume QSAR a hodnotení toxikologických vlastností chemických látok sa čoraz viac uplatňujú nové prístupy a nástroje, ktoré zlepšujú presnosť aj spoľahlivosť predikcií. V tejto kapitole sa budeme venovať prehľadu existujúcich riešení v oblasti QSAR výskumu, ako aj ich prínosom pre posudzovanie toxikologických vlastností chemických látok.

1.5.1 Ensemble learning

Medzi nedávne pokroky v QSAR pre predikciu podráždenia pokožky patrí štúdia Srisongkrama a kol. (2023) [38]. Výskum implementoval model učenia sa ensemble learning, ktorý využíva viaceré prediktívne algoritmy na zvýšenie presnosti a spoľahlivosti predikcií podráždenia pokožky. Tento model konkrétne využíval údaje o cytotoxicite odvodené z buniek HaCaT, ktoré sú široko používanou ľudskou keratinocytovou bunkovou líniou.

Prístup učenia sa *ensemble*, ktorý použili Srisongkram a kol., kombinoval rôzne jed-

notlivé modely, vrátane rozhodovacích stromov, SVM a neurónových sietí. Integráciou silných stránok týchto rôznorodých algoritmov bol model schopný zmierniť slabé stránky v jednotlivých modeloch.

1.5.2 Podráždenia pokožky hlbokých eutektických rozpúšťadiel

Výskum Li a kol. (2024) [21] skúmal aplikáciu QSAR modelov na štúdium potenciálu podráždenia pokožky hlbokých eutektických rozpúšťadiel (DES) v samozostavujúcich sa reverzných nanomiceľách. QSAR modely použité v tejto štúdii poskytli cenné poznatky o cytotoxických účinkoch týchto nanomieciel a uľahčili skúmanie rizík podráždenia pokožky spojených s transdermálnymi dodávacími systémami.

Hlboké eutektické rozpúšťadlá sú inovatívne rozpúšťadlá, ktoré v posledných rokoch sa využívajú kvôli svojej šetrnosti k životnému prostrediu a schopnosti tvoriť nanomicely. Reverzné nanomicele vytvorené z DES majú potenciál byť využívané v transdermálnych dodávacích systémoch, čo otvára nové možnosti v oblasti liečiv a kozmetiky. Avšak, ich bezpečnosť, najmä v súvislosti s podráždením pokožky, musí byť dôkladne preskúmaná.

Štúdia Li a kol. využila QSAR modely na predikciu cytotoxických a iritačných účinkov DES nanomieciel na pokožku. Tieto modely umožnili lepšie pochopiť, ako rôzne chemické zloženia a fyzikálno-chemické vlastnosti DES ovplyvňujú ich interakciu s kožnými bunkami. Výsledky výskumu ukázali, že QSAR modely môžu efektívne predpovedať potenciál podráždenia pokožky, čím poskytujú užitočné nástroje pre návrh bezpečných transdermálnych systémov.

1.5.3 QSAR Toolbox

QSAR Toolbox, ktorý využívali Anceschi a kol. (2023) [3], poskytuje platformu pre *in silico* testovanie chemických zlúčenín. Tento nástroj bol použitý na predikciu potenciálu podráždenia pokožky rôznych látok, čím umožnil rýchle a presné hodnotenia rizík.

QSAR Toolbox je softvérový nástroj, ktorý integruje rôzne metódy a algoritmy pre predikciu toxikologických vlastností chemických látok bez potreby fyzických testov na zvieratách alebo *in vitro* testov. Využitím veľkých databáz chemických, fyzikálno-chemických a biologických dát, QSAR Toolbox umožňuje vedcom a regulačným orgánom predpovedať potenciálne toxické účinky látok efektívne a nákladovo úsporne.

Štúdia Anceschi a kol. ukázala, že QSAR Toolbox môže úspešne predikovať podráždenie pokožky spôsobené rôznymi chemikáliami, čím podporuje rýchlejšie a presnejšie hodnotenie rizík v porovnaní s tradičnými metódami. Tento nástroj navyše pomáha pri plnení regulačných požiadaviek tým, že poskytuje spoľahlivé údaje pre rozhodovacie procesy v oblasti bezpečnosti chemických látok.

Kapitola 2

Návrh riešenia

Cieľom tejto kapitoly je podrobne popísať návrh aplikácie, ktorá slúži na predikciu dermatotoxicity chemických látok, a to so zameraním na iritáciu kože. Predikcia sa zakladá na metódach strojového učenia a kvantitatívnej analýze vzťahu medzi štruktúrou a biologickou aktivitou látok (QSAR). Táto kapitola zahŕňa špecifikáciu požiadaviek aplikácie, popis prípadov použitia, návrh pracovného postupu (workflow) a architektúru riešenia.

2.1 Špecifikácia požiadaviek

Stanovenie požiadaviek je dôležité pre efektívny návrh a implementáciu systému. Tieto požiadavky definujú očakávané schopnosti a vlastnosti aplikácie, ktoré sú nevyhnutné pre projekt. Požiadavky sú rozdelené do dvoch hlavných kategórií – funkcionálne a nefunkcionálne požiadavky. Funkcionálne požiadavky definujú konkrétne úlohy a schopnosti, ktoré systém musí vykonávať. Na druhej strane, nefunkcionálne požiadavky stanovujú kritériá kvality a technické parametre, ktoré musí aplikácia spĺňať.

2.1.1 Funkcionálne požiadavky

Funkcionálne požiadavky popisujú hlavné činnosti a vlastnosti aplikácie, ktoré zabezpečia splnenie jej účelu:

- **Načítanie datasetu:** Systém musí byť schopný načítať a spracovať vstupný dataset obsahujúci chemické látky a ich deskriptory.
- **Výpočet molekulárnych deskriptorov:** Systém musí vypočítať molekulárne deskriptory pre každú látku na základe jej SMILES kódu.
- **Predikcia iritácie kože:** Systém musí byť schopný predpovedať, či chemická látka spôsobí iritáciu kože, na základe vytrénovaného modelu.
- **Vizualizácia výsledkov:** Systém musí zobrazovať výsledky modelu a štatistiky výkonu modelu (napr. *accuracy*, *precision*, *recall*, F1 skóre).
- **Vysvetliteľnosť:** Systém musí byť schopný ukázať grafy vysvetliteľnosti, aby bolo možné vidieť na základe, čoho boli vykonané rozhodnutia.

2.1.2 Nefunkcionálne požiadavky

Nefunkcionálne požiadavky stanovujú technické kritériá a kvalitatívne parametre aplikácie:

- **Výkon:** Systém musí byť schopný spracovať a analyzovať aj väčšie množstvo chemických látok v primeranom čase.
- **Presnosť predikcie:** Systém musí zabezpečiť dostatočnú presnosť predikcie iritácie kože, pričom model musí byť schopný dosiahnuť minimálnu presnosť 80
- **Škálovateľnosť:** Systém musí byť škálovateľný a umožňovať spracovanie rôzne veľkých datasetov, vrátane možných budúcich rozšírení na viac látok

alebo vlastností.

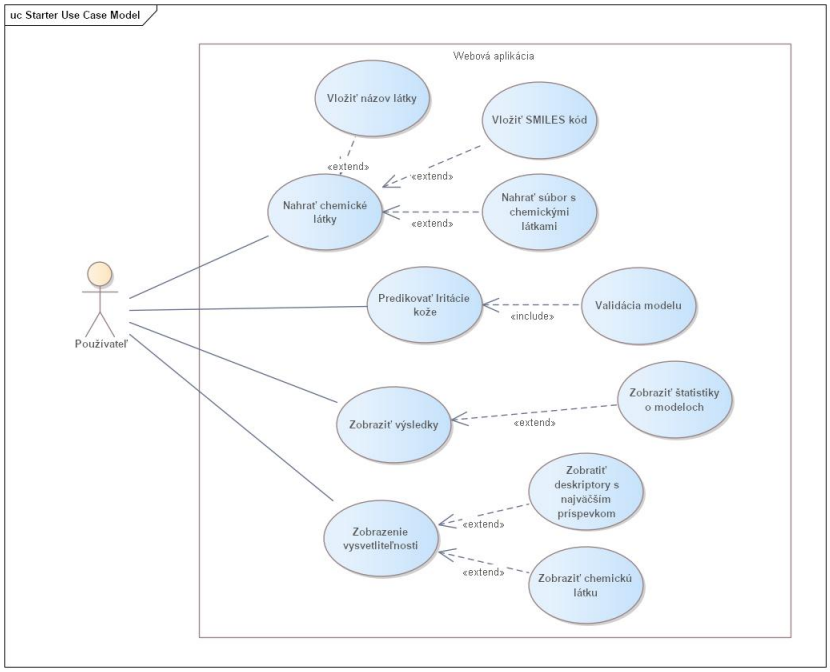
- **Spoľahlivosť:** Systém musí správne identifikovať neplatné a chýbajúce údaje a efektívne ich ošetriť, aby neovplyvňovali výsledky analýzy.

- **Použiteľnosť:** Systém musí byť používateľsky prívetivý, pričom interakcie s ním musia byť intuitívne aj pre používateľov bez technických znalostí.

- **Prispôsobiteľnosť:** Systém musí umožňovať úpravu a rozšírenie modelu, napríklad o nové deskriptory alebo iné predikčné úlohy.

2.2 Prípady použitia

Táto kapitola sa podrobnejšie venuje jednotlivým prípadom použitia aplikácie určenej na predikciu dermatotoxicity chemických látok. Každý prípad znázorňuje spôsob interakcie medzi používateľom a systémom v rôznych situáciách. Na základe definovaných požiadaviek a cieľových funkcionalít boli identifikované nasledujúce scenáre použitia:



Obr. 2.1: Diagram prípadov použitia.

2.2.1 Vloženie látky na zistenie predikcie

V tomto kroku používateľ zadáva chemickú látku, pre ktorú chce získať predikciu dermatotoxicity. Látka môže byť identifikovaná prostredníctvom SMILES zápisu alebo CAS čísla.

Kroky:

1. Zadanie CAS čísla alebo SMILES.
2. Systém skontroluje platnosť zadaného textu.
3. V prípade chyby je užívateľ informovaný o špecifickom probléme (napr. chýbajúce hodnoty).

2.2.2 Predspracovanie a výpočet deskriptorov

Systém automaticky spracováva nahranú látku, vypočíta molekulárne deskriptory a pripraví ich na vstup do modelu strojového učenia.

Kroky:

1. Užívateľ spustí proces predspracovania kliknutím na tlačidlo.
2. Systém transformuje SMILES kódy na molekulárne reprezentácie.
3. Na základe týchto reprezentácií sa vypočítavajú deskriptory.

2.2.3 Predikcia dermatotoxicity

Hlavný prípad použitia predstavuje predikciu, či chemická zlúčenina spôsobuje iritáciu pokožky. Systém analyzuje vstupy pomocou vopred vytrénovaného QSAR modelu a poskytuje užívateľovi predikcie.

Kroky:

1. Systém načíta predspracované dáta.
2. Model vykoná predikciu na základe vstupov.
3. Užívateľ dostane grafickú prezentáciu výsledkov predikcie

2.2.4 Vizualizácia a vysvetliteľnosť

Systém poskytuje vizualizácie, ktoré pomáhajú užívateľovi pochopiť rozhodovacie procesy modelu. Tieto vizualizácie zahŕňajú SHAP hodnoty, ktoré ukazujú vplyv jednotlivých deskriptorov alebo modelov na predikciu.

Kroky:

1. Systém analyzuje výsledky predikcie.

2. Generujú sa grafy vysvetliteľnosti.

2.3 Workflow

Táto kapitola popisuje pracovný postup trénovania modelov na predikciu dermatotoxicity chemických zlúčenín. Workflow pozostáva z niekoľkých krokov, ktoré zahŕňajú spracovanie dát, výpočet deskriptorov, trénovanie modelu a jeho validáciu. Jednotlivé kroky sú znázornené na obrázku.

Popis jednotlivých krokov:

1. **Zozbieranie dát:** Proces začína zbieraním dát, ktoré obsahujú chemické zlúčeniny reprezentované SMILES kódmi. Tieto dáta sú základom pre následnú analýzu a modelovanie.

2. **Načítanie dát:** Dáta sú nahraté do systému a automaticky validované. Overuje sa úplnosť a správnosť datasetu.

3. **Kontrola kvality dát:**

   - Ak dáta nie sú úplné, systém spustí **základné čistenie dát**.

   - V prípade nedostatku dát prebieha **augmentácia** alebo dodatočný zber dát.

4. **Výpočet deskriptorov:** Pre spracované SMILES kódy sa vypočítajú molekulárne deskriptory potrebné pre modelovanie. Tento krok zahŕňa:

   - Odstránenie chýb.

   - Normalizáciu dát.

5. **Výber deskriptorov:** Z celkového množstva vypočítaných deskriptorov sa vyberajú tie najrelevantnejšie na základe korelačnej analýzy a rozptylu hod-

nôt. Tie s vysokou koreláciou alebo nízkym rozptylom sú odstránené.

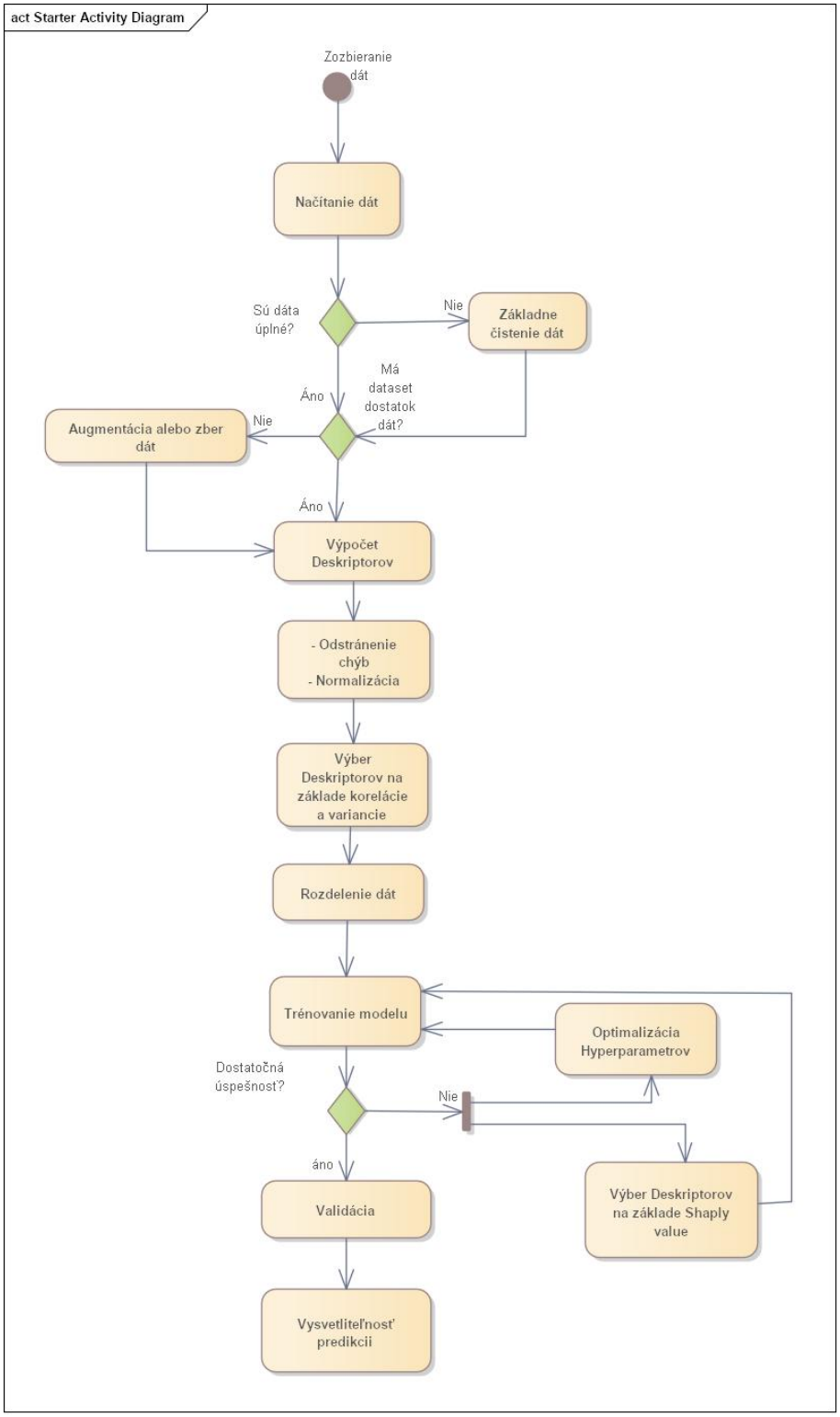
6. **Rozdelenie dát:** Dáta sa rozdelia na trénovaciu, testovaciu a validačnú množinu, aby bolo možné objektívne vyhodnotiť výkon modelu.

7. **Trénovanie modelu:** Systém trénuje modely strojového učenia na základe predspracovaných dát.

8. **Validácia:** Model je validovaný. Výkon modelu je analyzovaný na základe zvolených metrík (napr. úspešnosť predikcie).

9. **Optimalizácia hyperparametrov:** V prípade nedostatočnej úspešnosti sa vykonáva optimalizácia hyperparametrov modelu a ďalší výber deskriptorov pomocou metódy **SHAP (Shapley value)**.

10. **Vysvetliteľnosť predikcií:** Ak je úspešnosť modelu dostatočná, systém poskytuje grafickú prezentáciu výsledkov.



Obr. 2.2: Workflow diagram.

2.4 Architektúra riešenia

Aplikácia na predikciu dermatotoxicity bude postavená na architektúre Model-View-Controller (MVC). V rámci tejto architektúry sú jednotlivé komponenty rozdelené tak, aby boli funkčne nezávislé, čo prispieva k lepšej organizácii a udržateľnosti kódu. Nasledujúca časť popisuje hlavné logické komponenty aplikácie: Frontend a Backend.

2.4.1 Frontend (Užívateľské rozhranie)

Frontend aplikácie slúži ako užívateľské rozhranie, prostredníctvom ktorého môžu používatelia zadávať vstupy, spúšťať predikcie a zobrazovať výsledky. V tomto prípade je rozhranie navrhnuté tak, aby bolo intuitívne a prístupné aj pre používateľov bez technických znalostí. Užívateľ má možnosť vložiť chemické údaje (napr. v podobe SMILES kódov) pre látky, ktorých dermatotoxicitu chce posúdiť, spustiť predikciu dermatotoxicity, ktorá využíva backendové algoritmy strojového učenia na analýzu vstupných dát.

Následne poskytuje možnosť zobraziť výsledky predikcie spolu s vysvetlením jednotlivých deskriptorov, ktoré mali významný vplyv na výsledok predikcie. Frontend bude implementovaný s využitím HTML, CSS a JavaScriptu, aby poskytoval vhodné prostredie pre používateľov.

2.4.2 Backend (Spracovanie a predikcia)

Backend bude obsahovať jadrovú logiku aplikácie, ktorá zahŕňa predspracovanie dát, výpočet deskriptorov a predikciu dermatotoxicity. Nasledujúce komponenty budú zahrnuté v backendovej časti:

- Predspracovanie dát: Vstupné chemické údaje prechádzajú procesom čistenia

a transformácie na vhodné vstupy pre modely. Tento krok zahŕňa výpočet molekulárnych deskriptorov.

- Predikčné modely: V rámci backendu budú uložené modely strojového učenia, ktoré budú poskytovať predikcie dermatotoxicity.
- API rozhranie:Bude slúžiť na komunikáciu medzi frontendovou a backendovou časťou a na prijímanie požiadaviek od užívateľov.
- Vizualizácia vysvetlení: Backend bude generovať výstupy pre vysvetlenie predikcií, napríklad s využitím SHAP hodnôt, ktoré naznačujú, ktoré chemické deskriptory najviac ovplyvnili výsledok predikcie.

Backend je plne oddelený od užívateľského rozhrania, čo prispieva k modularite systému a umožňuje jeho lepšiu údržbu.

2.5 Výber metód strojového učenia

V tejto časti sa zaoberáme výberom algoritmov strojového učenia vhodných na vytvorenie prediktívneho modelu dermatotoxicity. Cieľom je zvoliť také algoritmy, ktoré poskytujú optimálnu kombináciu presnosti, efektivity a vysvetliteľnosti výsledkov, čím umožňujú vedecky podložené, ale zároveň používateľsky zrozumiteľné predikcie.

2.5.1 Decision tree (Rozhodovací strom)

Rozhodovací strom je algoritmus strojového učenia využívaný na klasifikáciu dát. Hlavnou výhodou rozhodovacích stromov je ich jednoduchá interpretácia – výstup modelu je možné vizualizovať ako sekvenciu pravidiel, čo uľahčuje vysvetlenie, prečo bola konkrétna látka klasifikovaná ako iritujúca alebo neiritujúca.

Rozhodovacie stromy sú zároveň vhodné na malé a homogénne datasety, pretože nevyžadujú veľké množstvo tréningových dát na vytvorenie spoľahlivého modelu. Ich nevýhodou však môže byť náchylnosť k pretrénovaniu, čo je možné čiastočne eliminovať použitím regularizácie [40].

2.5.2 Random Forest (Náhodný les)

Náhodný les je klasická *ensemble* metóda, ktorá kombinuje výsledky viacerých nezávisle trénovaných rozhodovacích stromov. V kontexte dermatotoxicity poskytuje Náhodný les spoľahlivý a stabilný výkon. Jeho hlavnou výhodou je odolnosť voči pretrénovaniu, čo znamená, že dokáže dobre generalizovať na nové dáta, čím sa znižuje riziko nesprávnych predikcií pri nových chemických zlúčeninách.

Náhodný les tiež ponúka vysvetliteľnosť výsledkov, a to prostredníctvom hodnotenia dôležitosti jednotlivých vstupných premenných. Vďaka tomu môžeme identifikovať kľúčové deskriptory, ktoré najviac ovplyvňujú dermatotoxicitu, a poskytnúť tak používateľom prehľad o relevantných chemických vlastnostiach [34, 31].

2.5.3 XGBoost (Extreme Gradient Boosting)

XGBoost je pokročilý model gradientného zosilňovania, ktorý kombinuje výkony viacerých rozhodovacích stromov za účelom maximalizácie presnosti predikcií. Tento algoritmus je v oblasti predikcie toxikologických vlastností často využívaný kvôli schopnosti prispôsobiť sa zložitým vzťahom medzi chemickými deskriptormi a cieľovými hodnotami, ako je dermatotoxicita.

Jednou z hlavných výhod XGBoostu je jeho schopnosť optimalizovať výkon pri vysokých nárokoch na presnosť a rýchlosť spracovania. Tento algoritmus je navrhnutý tak, aby minimalizoval chyby (napr. cez regularizáciu), pričom zachováva vysokú generalizačnú schopnosť. [12, 7, 46].

2.5.4 Support Vector Machine (SVM)

SVM je algoritmus často využívaný v klasifikačných úlohách, kde je cieľom rozdeliť dáta do presne definovaných kategórií. V tomto prípade je dermatotoxicita reprezentovaná ako binárna trieda (iritujúca vs. neiritujúca látka), čo robí SVM vhodným pre tento typ úlohy. Model sa sústredí na maximalizáciu rozdielu medzi triedami, čo pomáha dosahovať vysokú presnosť pri predikcii.

Pre úlohy toxikológie je SVM prínosný, pretože umožňuje dobré prispôsobenie modelu aj v prípadoch, keď je k dispozícii menší dataset. SVM tak zaručuje spoľahlivé výsledky aj pre menej známe alebo netypické zlúčeniny, čo prispieva k celkovej presnosti systému [18, 32].

2.5.5 KNN

KNN je jednoduchý, ale efektívny algoritmus strojového učenia, ktorý sa často využíva na klasifikačné úlohy. Jednou z hlavných výhod KNN je jeho schopnosť efektívne pracovať s malými datasetmi, čo je v kontexte QSAR modelov veľmi dôležitý. Homogénnosť dát (napr. numerické deskriptory chemických látok) zlepšuje jeho výkon, pretože algoritmus porovnáva hodnoty priamo v rámci rovnakého typu dát.

KNN má jednoduchú implementáciu a nevyžaduje zložitý tréningový proces, vďaka čomu je vhodný na dataset s obmedzeným počtom vzoriek. Nevýhodou však môže byť jeho závislosť na správnom nastavení parametra k (počet susedov) a na výbere metriky vzdialenosti [26, 48, 30].

Kapitola 3

Implementácia riešenia

V nasledujúcej kapitole je opísaná implementácia navrhnutej aplikácie na predikciu iritácie kože vo forme webovej aplikácie. Pre implementáciu bol zvolený framework Django kvôli jeho vstavaným komponentom a modulom, ktoré uľahčujú vývoj webových aplikácií.

Taktiež spĺňa požiadavky stanovené v návrhu systému, ako napríklad jeho architektúra je založená na model-view-controller, ktorá zabezpečuje oddelenie logiky aplikácie od prezentačnej vrstvy. To umožňuje lepšiu organizáciu kódu.

Dôležitú úlohu tiež zohráva verejne dostupná knižnica RDkit na výpočet deskriptorov.

3.1 Dataset

V rámci riešenia bol použitý dataset pozostávajúci z pôvodne 1703 záznamov chemických látok, z ktorých každá je charakterizovaná siedmimi atribútmi. Neskôr bol vrámci spracovania dát znížený počet na 703.

Jedným z hlavných atribútov je názov látky, ktorý je uvedený v stĺpci *Substance*. Tento názov slúži na identifikáciu chemickej zlúčeniny, čo je dôležité pre neskoršiu interpretáciu výsledkov.

Významným atribútom je aj CAS číslo (*CAS_number*), čo je jedinečný identifikátor priradený každej chemickej látke. Tento identifikátor je medzinárodne uznávaným štandardom, ktorý umožňuje jednoznačnú identifikáciu látok v chemickom priemysle a vo vedeckom výskume. CAS číslo zabezpečuje, že pri analýze je každá látka správne identifikovaná a spojená s presnými údajmi. [9]

Ďalším atribútom je aj SMILES kód (*Smiles_code*), ktorý reprezentuje chemickú štruktúru každej látky. SMILES je textový formát používaný na popis chemických zlúčenín a je široko používaný v chemických databázach. [33]

Dôležitým atribútom pre túto prácu je stĺpec *Irritation*, ktorý určuje reakcie látky s kožou. Látky v tomto datasete sú označené kategóriou I, čo znamená, že sú klasifikované ako iritujúce alebo NI, čo znamená neiritujúce. Všetky látky v datasete boli vyhodnotené na základe testovania na králikoch. Tento atribút bude použitý ako cieľová premenná v QSAR modeloch, kde model bude predikovať, či nová látka spôsobí iritáciu kože na základe jej chemických vlastností.

3.2 Výpočet deskriptorov

Dataset bol najskôr transformovavný na vhodné vstupy pre prediktívne modely. Na tento krok bola použitá knižnica RDKit, ktorá umožňuje výpočet molekulárnych deskriptorov na základe SMILES kódu.

Každý SMILES kód bol prevedený na molekulu, z ktorej boli následne vypočítané viaceré deskriptory, ako napríklad molekulová hmotnosť, počet atómov alebo polarizovateľnosť.

Ak sa pre niektorú látku nepodarilo vypočítať deskriptory, tieto hodnoty boli označené ako neplatné, aby sa predišlo chybám v neskoršom modelovaní.

3.3 Predspracovanie dát

V procese predspracovania údajov bola vykonaná séria krokov zameraných na úpravu a čistenie dát, aby boli vhodné na následné modelovanie a analýzu.

Po prvotnom výpočte molekulárnych deskriptorov bolo potrebné overiť kvalitu dát. V datasetoch často vznikajú neplatné hodnoty, buď kvôli neúplným záznamom alebo chybným výpočtom, a preto sa v rámci predspracovania vykonalo odstránenie záznamov, ktoré obsahovali chýbajúce alebo neplatné hodnoty. Tento krok bol nevyhnutný, aby sa predišlo chybám pri tréningu modelu a aby výsledné predikcie boli čo najpresnejšie.

Ďalším kľúčovým krokom bola identifikácia a analýza odľahlých hodnôt (*outlierov*). Odľahlé hodnoty môžu mať výrazný vplyv na tréning modelu, pretože ich prítomnosť môže skresľovať výsledky a ovplyvniť generalizáciu modelu na nové dáta. Na detekciu odľahlých hodnôt sa použili rôzne metriky na deskriptory, pričom odľahlé hodnoty, ktoré výrazne vybočovali z bežných rozsahov, boli analyzované a podľa potreby odstránené alebo ošetrené.

3.4 Implementácia Algoritmov

V tejto kapitole je opísaná konkrétna implementácia vybraných algoritmov strojového učenia, ktoré boli použité na vytvorenie finálneho predikčného modelu dermatotoxicity. Implementácia je realizovaná v Pythone s využitím knižníc xgboost, sklearn a numpy. Pred samotným tréningom modelov boli dáta rozdelené na trénovaciu a testovaciu množinu v pomere 80 ku 20.

Pri implementácii nastal problém, ktorý predstavoval pomerne malý dataset a tak úspešnosť pri spustení samotných modelov pred optimalizáciou hyperparametrov a vybratím menšieho počtu atribútov bola okolo 99%. Predovšetkým modely XGBoost, Náhodný les a Rozhodovací strom mali tento problém.

Na objektívnejšie posúdenie výkonu bola použitá k-folds validácia s k = 5, ktorá pri modeloch ukázala nasledujúcu presnosť predikcií:

Model	**Trénovanie**	**Testovanie**	**Krížová validácia**
XGBoost	0.993	0.837	0.790
Náhodný les	0.993	0.797	0.816
Rozhodovací strom	0.993	0.785	0.728
SVM	0.854	0.773	0.806
KNN	0.828	0.727	0.742

Tabuľka 3.1: Porovnanie presnosti rôznych modelov bez optimalizácie a výberu deskriptorov

3.5 Výber najvhodnejších deskriptorov

Po vypočítaní 1D a 2D deskriptorov bol ich počet 202. Vzhľadom na veľkosť datasetu a toho, že modely ukazovali známky pretrénovania sa ukázal byť tento počet príliš vysoký. Zníženie počtu deskriptorov v datasete pomôže modelom zamerať sa iba na tie, ktoré sú najviac dôležité pre predikciu iritácie kože. Ako ďalší krok bol teda zvolený výber najvhodnejších deskriptorov ako aj ich počet. Na túto úlohu boli otestované tri metódy - na základe korelácie a rozptylu, pomocou Lasso regresie a podľa SHAP hodnôt.

3.5.1 Výber na základe korelácie a rozptylu

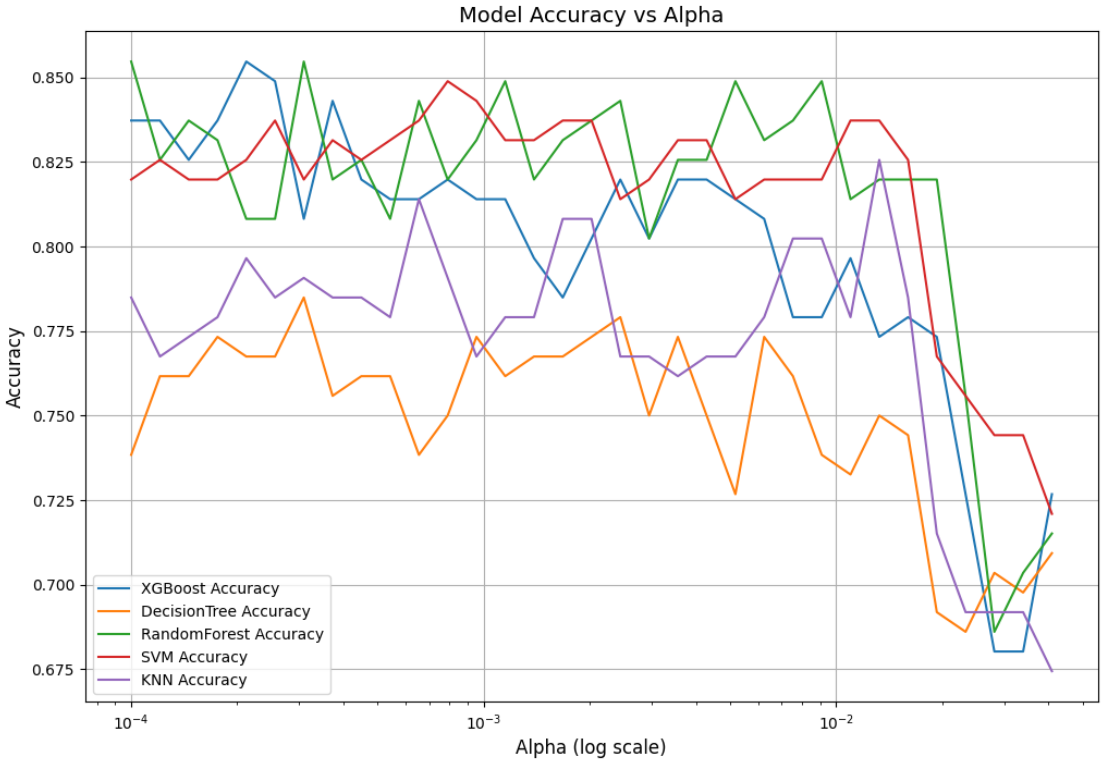
Najjednoduchším spôsobom je vylúčiť tie, ktoré sú úzko korelované alebo majú príliš nízky rozptyl. Môžeme ich odobrať z datasetu na základe toho, že vzorky s

vysokou koreláciou alebo nízkym rozptylom neprispievajú významne k celkovému výsledku, naopak môžu iba spôsobovať šum a nepresnosti.

Touto metódou sme znížili počet deskirptorov na 39, takže na pätinu, hoci vzhľadom na náš dataset to stále bolo priveľa.

3.5.2 Výber deskriptorov pomocou Lasso regresie

Ďalšou odtestovanou metódou bol výber deskriptorov pomocou Lasso regresie. Táto metóda bola zvolená z dôvodu jednoduchosti a rýchlosti výpočtu. Tiež zároveň zohľadňuje vzťah medzi deskriptormi a cieľovou premennou počas procesu výberu.



Obr. 3.1: Výber deskriptorov pomocou Lasso regresie.

Pri tejto metóde je najdôležitejšie optimálne nastaviť regularizačný parameter alfa.

V tejto práci bola použitá hodnota $\alpha = 0{,}005$. Táto hodnota bola vybratá na základe testovania viacerých možností, pričom cieľom bolo nájsť rovnováhu medzi počtom vybraných deskriptorov a úspešnosťou predikcii. Počas testovania sme sledovali vplyv α na presnosť, kde sa ukázalo, že v oblasti medzi 10^{-4} a 10^{-2} (t. j. medzi 0,0001 a 0,01) dosahovali modely najvyššie a najstabilnejšie hodnoty. Zvolili sme hodnotu $\alpha = 0{,}005$, ktorá sa nachádza práve v tejto stabilnej oblasti.

Príliš vysoká hodnota α by spôsobila príliš silnú penalizáciu a odstránila by aj mierne dôležité príznaky, čo by mohlo znížiť presnosť modelu. Naopak, príliš nízka hodnota by spôsobila, že Lasso by vybralo príliš veľké množstvo deskriptorov. Nastavenie $\alpha = 0{,}005$ sa teda ukázalo ako optimálne, keďže umožnilo vybrať primeraný počet vstupných premenných bez významnej straty dôležitých informácií.

Použitý postup bol nasledovný:

```
# Train a Lasso model
lasso = Lasso(alpha=0.005)
lasso.fit(X, y)


# Get important features (non-zero coefficients)
important_features = X.columns[np.abs(lasso.coef_) > 0]
print("Selected Features:", important_features)
```

3.5.3 Výber podľa SHAP hodnoty

Ďalšou metódou bol výber deskriptorov podľa SHAP hodnoty, čo je spôsob interpretácie výstupov modelov strojového učenia. Umožňuje určiť príspevok jednotlivých deskriptorov k predikcii, na základe čoho potom vieme vybrať tie najdôležitejšie a iba počet, ktorý dosahuje najlepšie výsledky.

Postup výberu deskriptorov na základe SHAP bol následovný:

1. **Tréning modelov:** Na začiatku boli modely strojového učenia trénované na datasete obsahujúcom všetky molekulárne deskriptory vypočítané z SMILES reprezentácií chemických zlúčení.

2. **Výpočet SHAP hodnôt:** Po natrénovaní modelov boli pre každý deskriptor vypočítané SHAP hodnoty, ktoré označujú príspevok konkrétneho deskriptoru k finálnej predikcii modelu. Tieto hodnoty odrážajú ako pozitívne aj negatívne ovplyvnenie predikcií.

3. **Agregácia a analýza:** Priemerná absolútna hodnota SHAP pre každý deskriptor bola použitá na určenie jeho globálnej dôležitosti. Týmto spôsobom boli identifikované najrelevantnejšie deskriptory.

4. **Vizualizácia:** Na interpretáciu výsledkov boli použité grafy, ktoré zobrazujú jednak globálnu dôležitosť deskriptorov, ako aj ich vplyv na jednotlivé predikcie modelu.

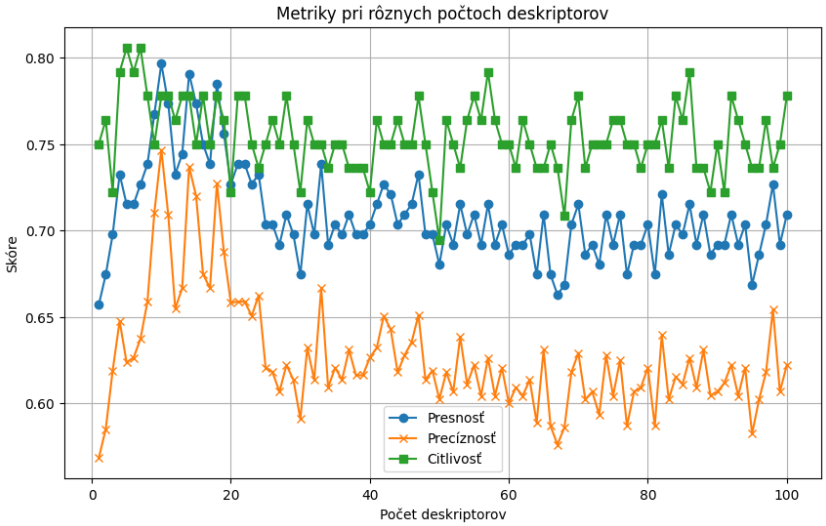
Počas experimentov boli vyskúšané rôzne počty deskriptorov po najvyššiu hodnotu 100, čo sme určili ako hornú hranicu. Potom bol vybratý počet deskriptorov, ktorý dosiahol najlepšie výsledky. Nakoľko boli modely stále pretrénované, proces sme opakovali dvadsaťkrát, aby sme si overili replikovateľnosť výsledkov a vybrali optimálny počet a aj tie deskriptory, ktoré sa vyskytli v čo najväčšej časti opakovaní.

3.5.3.1 Výber deskriptorov v Rozhodovacom strome

Rozhodovací strom je najstabilnejší a dosahuje najvyššie hodnoty metrík pri počte deskriptorov medzi 5 až 15. Pri tomto počte sa dosahovala najvyššia presnosť a zároveň bola vyvážená citlivosť a precíznosť. Naopak, pri väčšom počte deskrip-

torov dochádzalo k častému kolísaniu výkonu a miernemu poklesu presnosti, čo poukazuje na nadmernú komplexnosť modelu bez pridanej hodnoty.

Na základe týchto pozorovaní bol ako najvhodnejší zvolený počet deskriptorov 10. Táto hodnota predstavovala kompromis medzi výkonom modelu a jeho generalizačnou schopnosťou.



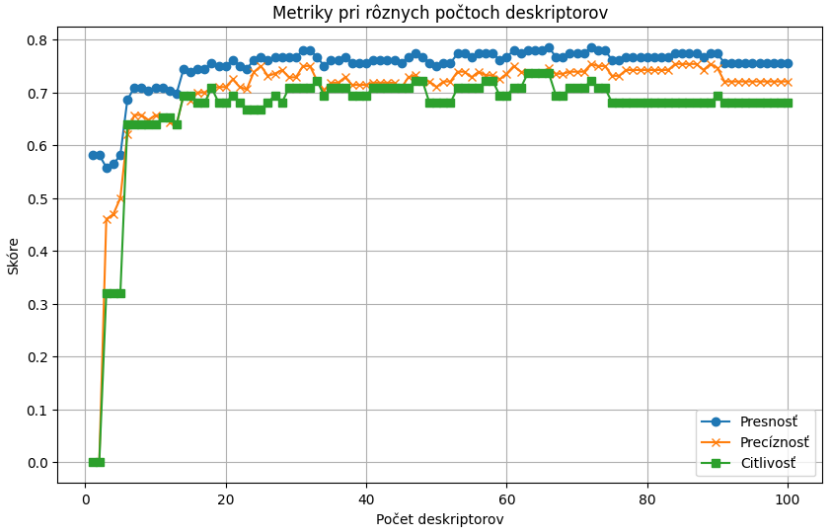
Obr. 3.2: Metriky pri rôznych deskriptoroch rozhodovacieho stromu.

3.5.3.2 Výber deskriptorov v SVM

Výkon modelu SVM je silne závislý od počtu použitých deskriptorov. Najvýraznejší nárast výkonu nastáva pri prvých 5 až 10 deskriptoroch, po ktorých sa hodnoty metrík stabilizujú. Presnosť modelu sa ustálila približne na úrovni 0,77, zatiaľ čo precíznosť dosahovala hodnoty okolo 0,73 a citlivosť sa pohybovala v rozmedzí 0,68 až 0,70. Pridávanie ďalších príznakov už neviedlo k výraznému zlepšeniu výkonu a v prípade citlivosti viedlo miestami dokonca k miernemu poklesu, čo môže naznačovať začínajúce pretrénovanie.

Model SVM, podobne ako Rozhodovací strom, potrebuje dostatočný počet des-

kriptorov na dosiahnutie spoľahlivých výsledkov. Na základe experimentov sme zvolili počet 9, kedy sa rast metrík ustálil a začal byť výkon stabilnejší.

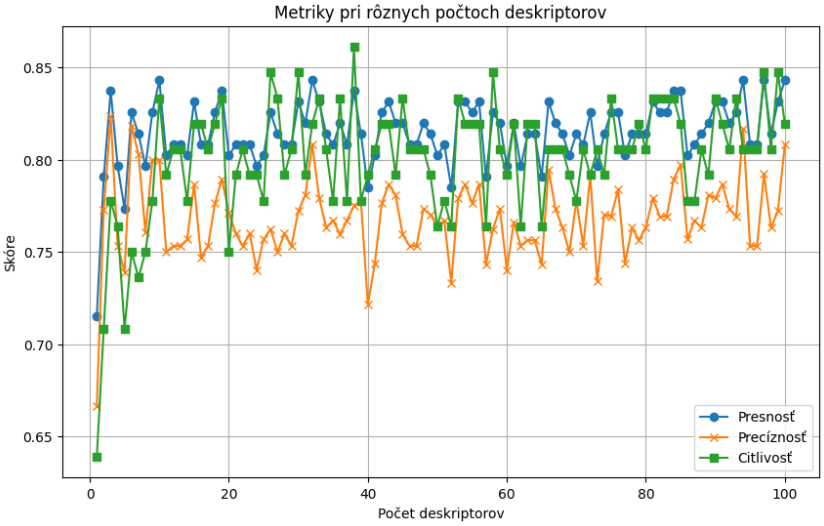


Obr. 3.3: Metriky pri rôznych deskriptoroch pre SVM.

3.5.3.3 Výber deskriptorov v Náhodnom lese

Náhodný les dosahoval vysokú a stabilnú presnosť už pri nízkom počte deskriptorov. Najvýraznejší nárast výkonu bol pozorovaný medzi 1 až 20 deskriptormi, pričom najlepšie výsledky sa opakovane objavovali v oblasti okolo 10 až 15 deskriptorov. Pri vyšších počtoch sa metriky síce výrazne nezhoršovali, avšak pribúdalo kolísanie, najmä pri precíznosti.

Na základe týchto pozorovaní bol ako optimálny zvolený počet 13 deskriptorov, pri ktorom model dosahoval najlepší kompromis medzi výkonom a jednoduchosťou.



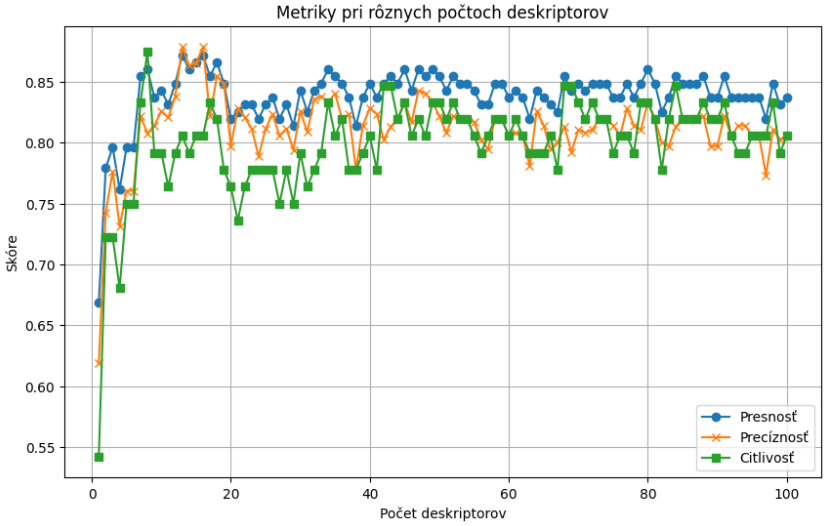
Obr. 3.4: Metriky pri rôznych deskriptoroch pre Náhodný les.

3.5.3.4 Výber deskriptorov v XGBoost

XGBoost dosahuje stabilne vysoké hodnoty metrík už pri relatívne nízkom počte deskriptorov. Najvýraznejší nárast vo výkone modelu (najmä v presnosti) bol zaznamenaný pri zvyšovaní počtu deskriptorov z 1 na približne 10.

Presnosť sa ustálila na hodnote okolo 0,82 až 0,83, čo predstavuje najvyššiu hodnotu spomedzi všetkých testovaných modelov. Precíznosť a citlivosť sa taktiež pohybovali v stabilných hodnotách okolo 0,80 až 0,83, čo ukazuje vyvážený výkon bez výrazného uprednostňovania jednej triedy.

Zároveň však možno pozorovať, že pri vyšších počtoch deskriptorov (nad 30) začína dochádzať k miernemu poklesu citlivosti a nárastu variability jednotlivých metrík. To naznačuje, že pridávanie ďalších vstupov už neprináša dodatočnú predikčnú hodnotu a môže dokonca znížiť schopnosť modelu generalizovať.



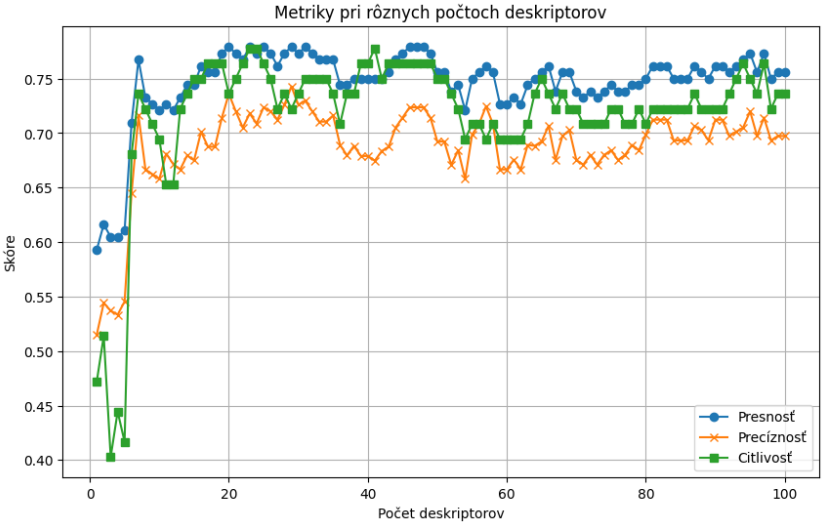
Obr. 3.5: Metriky pri rôznych deskriptoroch pre XGBoost.

Na základe týchto pozorovaní bol ako optimálny počet deskriptorov pre model XGBoost zvolený počet 12. Výber najdôležitejších 12 deskriptorov bol uskutočnený na základe agregovaných SHAP hodnôt, čo zabezpečilo, že do modelu boli zaradené len tie príznaky, ktoré mali najvýznamnejší vplyv na výslednú predikciu.

3.5.3.5 Výber deskriptorov v KNN

Výkon modelu KNN výrazne kolíše pri nízkom počte deskriptorov. Pri 1 až 10 deskriptoroch sú všetky metriky (presnosť, precíznosť, citlivosť) značne nestabilné, čo poukazuje na vysokú citlivosť modelu na výber vstupných premenných. Od približne 12. deskriptora sa hodnoty metrík začínajú stabilizovať.

Najvyššie skóre presnosti sa pohybuje okolo hodnoty 0.77 a bolo dosahované opakovane pri rôznych počtoch deskriptorov, najmä medzi 20 a 30. V tomto intervale boli zároveň metriky precíznosti a citlivosti najvyrovnanejšie. Precíznosť dosahovala hodnoty okolo 0.70 a citlivosť okolo 0.75, čo naznačuje dobrý kompromis medzi oboma typmi chýb (falošne pozitívne a falošne negatívne klasifikácie).



Obr. 3.6: Metriky pri rôznych deskriptoroch pre KNN.

Na základe týchto pozorovaní bol ako optimálny počet deskriptorov pre model KNN zvolený počet 22. Tento výber zabezpečuje nielen dobrý predikčný výkon, ale aj dostatočnú stabilitu metrík naprieč viacerými testovanými variáciami modelu. Deskriptory boli vybrané na základe ich prínosu podľa agregovaných SHAP hodnôt.

3.5.4 Zhodnotenie výberu deskriptorov

Pri porovnaní troch použitých metód výberu deskriptorov – korelácie a rozptylu, LASSO a SHAP – je zrejmé, že každá z nich mala odlišný dopad na výkon jednotlivých modelov.

Najmenej presný a najmenej stabilný výkon bol pozorovaný pri použití deskriptorov vybraných metódou korelácie a rozptylu. Hoci táto metóda výrazne znížila počet príznakov (zo 202 na 39), nevyužíva informáciu o vzťahu medzi príznakmi a cieľovou premennou, čo môže viesť k zachovaniu menej informatívnych znakov a vylúčeniu užitočných.

Tabuľka 3.2: Výsledky modelov po výbere príznakov pomocou korelácie a rozptylu

Model	**Tréning**	**Test**	**Precíznosť**	**Citlivosť**	**F1**	**CV**
XGBoost	0.9956	0.7791	0.7179	0.7778	0.7467	0.8029
Náhodný les	0.9956	0.7907	0.7432	0.7639	0.7534	0.8190
Rozhodovací strom	0.9956	0.7093	0.6528	0.6528	0.6528	0.7445
KNN	0.8307	0.7093	0.6170	0.8056	0.6988	0.7489
SVM	0.8467	0.7267	0.6623	0.7083	0.6846	0.7723

Naopak, LASSO regresia aj SHAP hodnoty umožňujú priamu selekciu najvýznamnejších príznakov na základe ich vplyvu na predikciu. V prípade LASSO sa zlepšila testovacia presnosť vo všetkých modeloch, pričom najvyššiu presnosť dosiahol model Náhodný les (test acc. 0.8198, F1 = 0.7801).

Tabuľka 3.3: Výsledky modelov po výbere príznakov pomocou LASSO

Model	**Tréning**	**Test**	**Precíznosť**	**Citlivosť**	**F1**	**CV**
XGBoost	0.9956	0.7965	0.7403	0.7917	0.7651	0.8000
Náhodný les	0.9956	0.8198	0.7971	0.7639	0.7801	0.8204
Rozhodovací strom	0.9956	0.7681	0.7671	0.7778	0.7724	0.7401
KNN	0.8292	0.7500	0.6933	0.7222	0.7075	0.7664
SVM	0.7898	0.7733	0.7089	0.7778	0.7417	0.7752

SHAP hodnoty umožnili ešte presnejšiu selekciu, keďže sme mohli individuálne vybrať optimálny počet deskriptorov pre každý model zvlášť. Výsledky ukázali, že pre niektoré modely (napr. KNN alebo SVM) bolo vhodné zvoliť vyšší počet deskriptorov (20–22), zatiaľ čo pre iné (napr. Rozhodovací strom alebo XGBoost) stačilo menej ako 15. Najvyššie celkové metriky boli dosiahnuté pri použití SHAP výberu v modeli XGBoost (presnosť až 0.85), ktorý mal aj najlepšiu rovnováhu medzi citlivosťou a precíznosťou.

Tabuľka 3.4: Výsledky modelov po výbere príznakov pomocou SHAP

Model	**Tréning**	**Test**	**Precíznosť**	**Citlivosť**	**F1**	**CV**
XGBoost	0.9956	0.8140	0.7632	0.8056	0.7838	0.8282
Náhodný les	0.9927	0.8140	0.7381	0.8611	0.7949	0.8102
Rozhodovací strom	0.9927	0.8081	0.8000	0.7222	0.7591	0.7562
KNN	0.8102	0.7907	0.7195	0.8194	0.7662	0.7533
SVM	0.7985	0.7849	0.7536	0.7222	0.7376	0.7854

Vo všetkých prípadoch však bola trénovacia presnosť (*train accuracy*) výrazne vyššia než testovacia, často presahovala 99 %. Toto je znakom pretrénovania (*overfitting*), najmä pri modeloch ako Rozhodovací strom alebo Náhodný les. Preto sme sa pri vyhodnocovaní nezameriavali len na testovaciu presnosť, ale zohľadnili sme aj hodnoty krížovej validácie (*cross-validation accuracy*), ktoré lepšie odrážajú schopnosť modelu generalizovať na nové dáta.

3.6 Optimalizácia

V tejto kapitole sa zameriame na optimalizáciu hyperparametrov jednotlivých modelov. Každý model vyžaduje úpravu špecifických parametrov, ktoré ovplyvňujú jeho výkon. Optimalizáciou týchto parametrov sme sa snažili dosiahnuť predovšetkým zníženie pretrénovania.

Pri optimalizácii hyperparametrov modelu bol použitý nástroj Optuna, ktorý slúži na hľadanie optimálnych nastavení modelov strojového učenia. Optuna je knižnica navrhnutá pre adaptívnu optimalizáciu hyperparametrov, čo znamená, že dokáže inteligentne prehľadávať možné hodnoty hyperparametrov tak, aby minimalizovala počet potrebných experimentov a zároveň dosiahla čo najlepšie výsledky.

V nasledujúcich podkapitolách sa budeme venovať optimalizácii pre konkrétne modely a vysvetlíme, ktoré parametre boli optimalizované a prečo.

3.6.1 XGBoost (Extreme Gradient Boosting)

Pre optimalizáciu modelu XGBoost boli vybrané nasledujúce najdôležitejšie hyperparametre:

- `n_estimators`: Počet stromov v modeli. Tento parameter určuje, koľko rozhodovacích stromov bude model generovať. Vyšší počet stromov môže zlepšiť výkon, ale zároveň zvyšuje výpočtovú náročnosť.
- `max_depth`: Maximálna hĺbka jednotlivých stromov. Tento parameter kontroluje komplexnosť jednotlivých stromov. Nižšia hodnota znamená menej komplexné stromy, zatiaľ čo vyššia hodnota môže viesť k modelu s vyššou variabilitou, ktorý sa môže pretrénovať.
- `learning_rate`: Rýchlosť učenia, ktorá určuje, aký veľký krok robí model pri učení. Príliš vysoká hodnota môže viesť k pretrénovaniu modelu, zatiaľ čo nízka hodnota môže spomaliť proces učenia, ale zvyčajne vedie k lepšiemu výkonu.
- `subsample`: Podiel vzoriek použitých na tréning. Tento parameter je dôležitý pre reguláciu modelu, pretože znižuje riziko pretrénovania.
- `colsample_bytree`: Podiel vstupných atribútov, ktoré sa použijú na každý rozhodovací strom. Tento parameter ovplyvňuje, ako veľmi bude model závislý na jednotlivých atribútoch.
- `gamma`,`alpha`,`lambda`: Regularizačné parametre, ktoré slúžia na kontrolu komplexnosti modelu a predchádzanie pretrénovaniu.

Väčšinu nastavení ako napríklad max_depth v rozsahu 3-10 alebo learning_rate v rozsahu 0.01-0.3, sme zvolili na základe všeobecných odporúčaní a predchádzajúcich experimentoch s XGBoost. Tieto hodnoty sa ukázali byť efektívne pre rôzne

typy problémov, takže sa často používajú na začiatku optimalizovania. Pri výbere týchto hodnôt sme zohľadnili minimalizovanie pretrénovania. Parametre ako max_depth, subsample a colsample_bytree môžu výrazne ovplyvniť, ako veľmi model generalizuje na nevidených dátach.

Každá kombinácia parametrov bola vyhodnotená pomocou krížovej validácie (cross-validation), pričom ako hodnotiaci parameter bol použitý F1 skóre a presnosť.

Na základe výsledkov optimalizácie sme našli najlepšie hodnoty pre každý parameter pre tento konkrétny dataset:

- `n_estimators`: 82
- `max_depth`: 5
- `learning_rate`: 0.1
- `subsample`: 0.65
- `colsample_bytree`: 0.76
- `min_child_weight`: 8
- gamma : 0.89
- lambda : 20.12
- alpha : 6.15

Tieto hodnoty poskytli najlepší výkon v porovnaní s inými kombináciami hyperparametrov.

3.6.2 Rozhodovací strom (Decision Tree)

Rozhodovacie stromy sú jedným zo základných algoritmov strojového učenia a preto môžu byť náchylné k pretrénovaniu, najmä ak strom rastie príliš hlboko.

Preto je veľmi dôležité optimalizovať jeho hyperparametre, aby sme dosiahli vyvážený model. Hlavné parametre, ktoré sme optimalizovali pre rozhodovací strom, sú:

- `max_depth`: Určuje maximálnu hĺbku stromu. Hĺbka stromu určuje, ako hlboko bude strom rozdelený.
- `min_samples_split`: Určuje minimálny počet vzoriek, ktoré musia byť prítomné v uzle, aby došlo k rozdeleniu. Tento parameter pomáha kontrolovať, ako veľké a rozmanité môžu byť uzly v strome. Vyššia hodnota môže zabrániť príliš detailnému rozdeľovaniu.
- `min_samples_leaf`: Určuje minimálny počet vzoriek, ktoré musia byť v listovom uzle. Tento parameter zabezpečuje, že uzly na konci stromu nebudú obsahovať príliš málo dát, čím sa zníži riziko pretrénovania.
- `max_features`: Určuje maximálny počet príznakov, ktoré sa budú používať na rozhodovanie v každom uzle stromu. Tento parameter môže pomôcť zlepšiť schopnosť modelu generalizovať tým, že zníži riziko toho, že sa model prispôsobí špecifickým šumom v dátach.
- `criterion`: Určuje, akú metódu použije model na hodnotenie kvality rozdelení uzlov. Bežné možnosti sú `gini` (Gini index) a `entropy` (Shannonova entropia). Gini index je rýchlejší a bežne používaný v praxi, zatiaľ čo entropia je zvyčajne presnejšia, ale môže byť náročnejšia na výpočty.

V tomto prípade sme sa zamerali na optimalizáciu hĺbky stromu (`max_depth`), ktorá je jedným z najdôležitejších parametrov rozhodovacieho stromu. Zároveň sme testovali rôzne hodnoty `min_samples_split` a `min_samples_leaf`, aby sme zabránili príliš hlbokému alebo príliš plytkému stromu, čím sa minimalizoval problém s pretrénovaním. Parametre `max_features` a `criterion` boli testované s cieľom

nájsť najlepšie nastavenia pre naše konkrétne dáta.

Optimálne parametre:

- {`criterion`: `gini`,
- `max_depth`: 5,
- `min_samples_split`: 19,
- `min_samples_leaf`: 15,
- `max_features`: `None`,
- `splitter`: `best`}

Po optimalizácii pomocou Optuny sme získali najlepšiu kombináciu hyperparametrov, ktorá zlepšila výkon modelu. Tieto optimalizované parametre zabezpečili lepšiu schopnosť modelu generalizovať na nových dátach a zlepšili jeho presnosť.

3.6.3 Náhodný les (Random forest)

Optimalizácia hyperparametrov modelu Náhodný les sa zameriava na niekoľko hlavných parametrov, ktoré ovplyvňujú výkon modelu.

- `n_estimators`: Počet stromov v lese. Tento parameter určuje, koľko rozhodovacích stromov bude model obsahovať.
- `max_depth`: Maximálna hĺbka jednotlivých stromov. Tento parameter kontroluje zložitosť stromu.
- `min_samples_split`: Minimálny počet vzoriek potrebných na rozdelenie uzla.
- `min_samples_leaf`: Minimálny počet vzoriek v liste.
- `max_features`: Počet príznakov, ktoré budú použité na vytvorenie každého

rozhodovacieho stromu.

Tieto parametre boli vybrané na základe odporúčaní a štandardných hodnôt používaných v praxi. Pri optimálnom nastavení parametrov model Náhodný les dosahuje dobrú rovnováhu medzi biasom a varianciou, čo vedie k vyššiemu výkonu na neznámych dátach a zabraňuje pretrénovaniu.

Optimálne parametre:

- {`n_estimators`: 168,
- `max_depth`: 5,
- `min_samples_split`: 25,
- `min_samples_leaf`: 5,
- `max_features`: `sqrt`,
- `bootstrap`: `True`}

3.6.4 SVM (Support Vector Machine)

Pre optimalizáciu modelu SVM bolo potrebné nastaviť niekoľko hyperparametrov, ktoré môžu výrazne ovplyvniť výkon modelu. Parametre, ktoré boli použité pri optimalizácii modelu SVM, sú nasledovné:

- `C`: Tento parameter určuje penalizáciu pre chyby modelu. Vyššia hodnota `C` znamená, že model bude menej tolerovať chyby na trénovacích dátach (to môže viesť k overfittingu), zatiaľ čo nižšia hodnota `C` môže zlepšiť generalizovateľnosť modelu, ale zároveň môže zvýšiť počet chýb na trénovacích dátach.
- `kernel`: Tento parameter určuje typ jadra, ktorý sa použije pri vytváraní

rozhodovacej hyperroviny. Bežné možnosti sú `linear`, `poly`, `rbf` (radial basis function) a `sigmoid`.

- `gamma`: Parameter `gamma` určuje, aký veľký je vplyv jednotlivých vzoriek na rozhodovaciu hranicu. Vyššie hodnoty znamenajú, že jednotlivé vzorky majú väčší vplyv na model, čo môže viesť k pretrénovaniu. Naopak, nižšie hodnoty môžu zjednodušiť model a zlepšiť jeho schopnosť generalizovať.

- `degree`: Tento parameter je relevantný iba pre `poly` jadro a určuje stupeň polynómu, ktorý sa používa pri transformácii dát. Tento parameter bol optimalizovaný iba v prípade, že sa ako kernel použil polynóm. `rbf` kernel.

V tomto prípade bolo dôležité otestovať rôzne jadrá a hodnoty parametra `C`, aby sa zistilo, ako veľmi penalizovanie chýb ovplyvňuje výkon modelu. Taktiež sa testovali rôzne hodnoty `gamma` s cieľom nájsť optimálnu hodnotu, ktorá by pomohla modelu nájsť optimálne rozhodovacie hranice medzi triedami. Hlavným cieľom bolo nájsť kombináciu týchto parametrov, ktorá by minimalizovala chyby.

Optimálne parametre:

- `kernel`: `rbf`,
- `C`: 8.90,
- `gamma`: `scale`

3.6.5 K-Nearest Neighbors (KNN)

Pri optimalizácii modelu K-Nearest Neighbors (KNN) bolo taktiež potrebné nastaviť niekoľko hyperparametrov, ktoré mali zásadný vplyv na jeho výkon. Tento model, ktorý sa zakladá na princípe vyhľadávania najbližších susedov pre každú predikciu, vyžaduje nastavenie parametrov, ktoré určujú, ako sa tieto susedné body

vyberajú a aký vplyv majú na konečný výsledok.

Hlavné parametre, ktoré sme optimalizovali pri modelovaní KNN, sú:

- `n_neighbors`: Tento parameter určuje počet najbližších susedov, ktorí budú použité na klasifikáciu alebo regresiu. Vyšší počet susedov môže viesť k vyššej generalizovateľnosti modelu, ale môže tiež znížiť presnosť, ak je dátová distribúcia veľmi heterogénna.

- `weights`: Tento parameter určuje, akým spôsobom bude priradená váha k susedom pri predikcii. Možnosti sú `uniform`, kde všetci susedia majú rovnakú váhu, a `distance`, kde váha susedov závisí od ich vzdialenosti od bodu, pre ktorý sa robí predikcia.

- `algorithm`: Tento parameter určuje, aký algoritmus sa použije na vyhľadávanie susedov. Bežné možnosti sú `ball_tree`, `kd_tree` a `brute`, kde `ball_tree` a `kd_tree` sú efektívne pre veľké dataset, zatiaľ čo `brute` je jednoduchý, ale pomalý, preto sa používa hlavne pri menších datasetoch.

- `metric`: Určuje metriku, ktorá sa používa na výpočet vzdialenosti medzi bodmi. Bežné možnosti sú `euclidean`, `manhattan`, `minkowski` a ďalšie. Zvolená metrika môže mať výrazný vplyv na výsledky modelu.

V tomto prípade sme sa rozhodli testovať rôzne hodnoty parametra `n_neighbors`, aby sme zistili, ako počet susedov ovplyvňuje presnosť modelu. Zároveň sme sa zamerali na rôzne možnosti váh (`uniform` a `distance`), aby sme našli najlepšiu metódu, ktorá by dokázala lepšie odrážať dôležitosť jednotlivých susedov na výsledok predikcie.

Optimálne parametre:

- `n_neighbors`: 5,

- `weights`: `distance`,
- `algorithm`: `kd_tree`,
- `p`: 2

Po optimalizácii parametrov KNN sme dosiahli stabilné zlepšenie presnosti modelu a jeho schopnosť správne klasifikovať nové, neznáme vzorky sa výrazne zvýšila.

3.6.6 Výsledky optimalizácie hyperparametrov

Najlepšie výsledky naprieč všetkými metrikami dosiahol model XGBoost, ktorý sa vyznačoval najvyššou presnosťou (0.84) a vyváženým pomerom citlivosti a precíznosti. Nasledoval Náhodný les, ktorý taktiež vykazoval stabilné výsledky a dostatočnú generalizáciu.

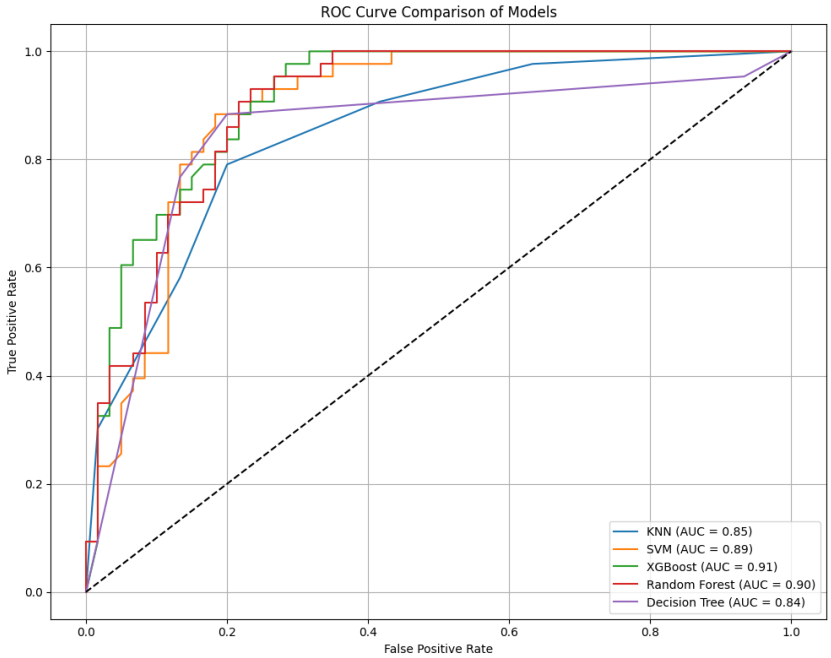
Naopak, model Rozhodovací strom, aj keď jednoduchý a rýchly na trénovanie, mal najnižšiu presnosť (0.75) a bol výrazne náchylný na pretrénovanie. Modely SVM a KNN poskytli stredne dobré výsledky, pričom KNN zaznamenal mierne lepšiu citlivosť, no nižšiu precíznosť.

Model	**Presnosť**	**Precíznosť**	**Citlivosť**	**F1-Skóre**
XGBoost	0.84	0.82	0.86	0.84
Náhodný les	0.82	0.78	0.82	0.80
Rozhodovací strom	0.75	0.74	0.77	0.75
SVM	0.78	0.75	0.78	0.76
KNN	0.79	0.76	0.79	0.77

Tabuľka 3.5: Výsledky modelu po optimalizácii a výbere deskriptorov

Najvyššiu plochu pod krivkou dosiahol model XGBoost, s hodnotou AUC = 0.91. Krivka sa drží nad diagonálou náhodnej klasifikácie, pričom už pri nízkej miere falošných pozitív dosahuje vysokú mieru správnych pozitív.

Ostatné modely mali tiež podobné výsledky, pričom Rozhodovací strom mal najmenšiu hodnotu AUC a spolu aj s KNN najjednoduchšiu krivku kvôli jeho malej hĺbke a malému počtu susedov u KNN.



Obr. 3.7: AUC ROC krivky pre jednotlivé modely.

Výsledky ukazujú, že optimalizácia hyperparametrov mala výrazný vplyv na výkon jednotlivých modelov. V prípade pokročilých algoritmov ako XGBoost a Náhodný les optimalizácia viedla k výraznému zlepšeniu všetkých hodnotených metrík.

Z hľadiska praktického využitia sa preto ako najvhodnejší model ukázal byť XGBoost, ktorý zároveň poskytuje aj dobrú interpretovateľnosť prostredníctvom SHAP analýzy.

3.7 Optimalizácia datasetu

Po úvodnej fáze optimalizácie hyperparametrov modelov bolo zistené, že aj napriek rozsiahlemu ladeniu parametrov a posunutiu rozhodovacej hranice, sa nepodarilo výrazne zlepšiť kľúčové metriky ako precíznost, citlivosť a F1 skóre. Modely mali tendenciu uprednostňovať väčšinovú triedu (neiritujúce látky), čo spôsobovalo výrazné zníženie schopnosti správne identifikovať menšinovú triedu hoci modely dosahovali postačujúcu presnosť.

Na základe tejto analýzy bola spravená úprava samotného datasetu. Hlavným cieľom bolo zabezpečiť, aby modely mali k dispozícii reprezentatívnejšiu a vyváženejšiu vzorku dát.

3.7.1 Zníženie počtu vzoriek pomocou klastrovania

Keďže pôvodný dataset obsahoval približne štvornásobne viac vzoriek neiritujúcich látok v porovnaní s iritujúcimi, bola navrhnutá metóda výberu reprezentatívnych vzoriek pomocou klastrovania. Konkrétne bol použitý algoritmus k-means s preddefinovaným počtom klastrov. Pre každý klaster bola vybraná vzorka najbližšia k centroidu, čím sa podarilo znížiť počet vzoriek väčšinovej triedy bez straty diverzity dát.

Takto vybraný súbor vzoriek bol následne spojený so všetkými dostupnými vzorkami minoritnej triedy a výsledný dataset bol náhodne premiešaný.

3.7.2 Výsledky po optimalizácii datasetu

Porovnanie výkonnosti modelov ukázalo výrazné zlepšenie v metrikách citlivosť, precíznosť a F1 skóre naprieč všetkými modelmi.

Tieto výsledky potvrdzujú, že samotná optimalizácia hyperparametrov bez úpravy

Tabuľka 3.6: Výsledky modelov na not balanced datasete

Model	**Presnosť**	**Precíznosť**	**Citlivosť**	**F1 Skóre**
XGBoost	0.820	0.607	0.685	0.643
Náhodný les	0.837	0.676	0.352	0.463
Rozhodovací strom	0.820	0.569	0.408	0.475
SVM	0.846	0.750	0.338	0.466
KNN	0.840	0.625	0.493	0.551

Tabuľka 3.7: Výsledky modelov na vyváženom datasete

Model	**Presnosť**	**Precíznosť**	**Citlivosť**	**F1 Skóre**
XGBoost	0.826	0.730	0.852	0.786
Náhodny les	0.820	0.766	0.819	0.792
Rozhodovací strom	0.773	0.762	0.667	0.711
SVM	0.791	0.757	0.736	0.746
KNN	0.801	0.709	0.847	0.772

dátového súboru nebola postačujúca na riešenie problému nevyváženosti tried.

3.8 Spojenie modelov

Na zlepšenie predikčných výsledkov, ktoré pri jednotlivých modeloch kolísali v závislosti od konkrétneho rozdelenia dát medzi trénovacou a testovacou množinou, boli implementované *ensemble* prístupy. Okrem toho dôvodom bolo aj zlepšiť správanie modelu z pohľadu metrík ako sú presnosť, citlivosť alebo F1 skóre, keďže kombinujú silné stránky jednotlivých algoritmov. V tejto práci boli využité dve metódy: spojenie pomocou (*metamodelu*) a hlasovanie (*voting*).

3.8.1 Spojenie pomocou metamodelu

Prvá implementovaná metóda bola založená na spojení modelov pomocou logistickej regresie ako meta-modelu. Použité boli doteraz samostatne natrénované modely s ich jednotlivými deskriptormi zvolenými na základe SHAP hodnôt.

Predikcie týchto základných modelov boli následne použité ako vstupné premenné pre meta-model – logistickú regresiu. Pre overenie, či modely prispievajú štatisticky významne ku predikcii sme si nechali vypísať ich koeficienty. Významnosť jednotlivých modelov v meta-modeli je uvedená v tabuľke:

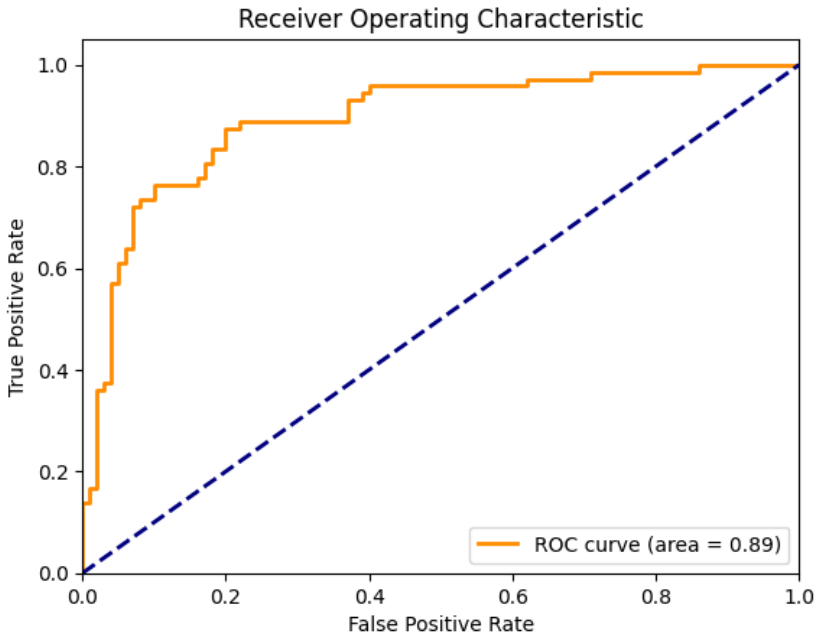
Tabuľka 3.8: Koeficienty modelov v stacking ensemble (logistická regresia)

Model	**Koeficient**
XGBoost	1.491
Rozhodovací strom	1.209
Náhodný les	1.043
SVM	1.017
KNN	0.213

Na základe toho môžeme vidieť, že v porovnaní s ostatnými KNN má nízky koeficient, a teda významne neprispieva, preto bol tento model vynechaný a logistická regresia kombinuje predikcie modelov XGBoost, Rozhodovací strom, Náhodný les a SVM. Finálne metriky sú v nasledujúcej tabuľke:

Tabuľka 3.9: Výkon stacking modelu

Metrika	**Hodnota**
Testovacia Presnosť	0.878
Precíznosť	0.807
Citlivosť	0.931
F1 Skóre	0.865



Obr. 3.8: Stacing model AUC ROC krivka.

3.8.2 Hlasovanie

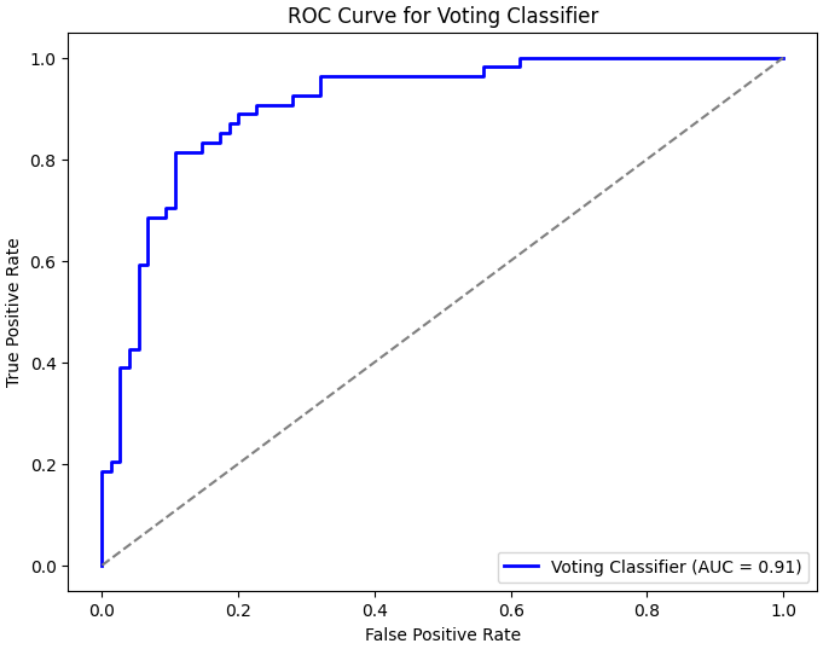
Druhá z použitých metód bolo spojenie modelov pomocou hlasovania. V tomto prípade bolo nevyhnutné, aby všetky modely boli trénované na rovnakej množine deskriptorov. Na tento účel boli použité vybrané deskriptory pomocou Lasso regularizácie, keďže dosahovali dobré výsledky a pre všetky modely boli rovnaké.

Bolo použité tvrdé hlasovanie (*hard voting*) boli použité všetky dostupné modely. Tento prístup funguje na princípe väčšinového rozhodovania, kde konečná predikcia zodpovedá najčastejšiemu výstupu jednotlivých klasifikátorov.

Dosiahnuté výsledky sú uvedené v tabuľke:

Tabuľka 3.10: Výkon voting ensemble modelov

Typ hlasovania	Presnosť	Precíznosť	Citlivosť	F1
Hard Voting	0.82	0.79	0.83	0.81
Soft Voting	0.83	0.80	0.85	0.82



Obr. 3.9: Hlasovací model AUC ROC krivka.

3.9 Webové rozhranie aplikácie

Na využívanie modelu bola navrhnutá a implementovaná webová aplikácia, ktorá umožňuje používateľom jednoducho interagovať so systémom a získať predikcie na základe zadanej chemickej zlúčeniny. Vybratý model bol model, ktorý kombinuje výstupy pomocou logistickej regresie. Aplikácia je postavená na klient-server architektúre, kde frontendová časť využíva `React` (v rámci HTML šablóny s JavaScriptom) a backend je implementovaný pomocou Python frameworku `Django`.

3.9.1 Backendová časť aplikácie

Backendová časť aplikácie je implementovaná pomocou webového frameworku Django, ktorý umožňuje jednoduché spracovanie HTTP požiadaviek, routing a integráciu s databázou, ak by to bolo potrebné. Hlavná funkcionalita je v súbore `views.py`, kde sa nachádza pohľad `predict`, ktorý reaguje na POST požiadavky odosielané z používateľského rozhrania.

Tento pohľad prijíma vstup v podobe SMILES reťazca alebo CAS čísla. Po prijatí požiadavky sa vstup najprv analyzuje pomocou pomocnej funkcie `process_input`, ktorá zabezpečuje detekciu typu vstupu, jeho prípadný preklad (napr. CAS čísla na SMILES) a výpočet molekulových deskriptorov pomocou knižnice RDKit. Výsledkom je číselný vektor s deskriptormi zoradený podľa vstupného poradia používaného pri trénovaní modelu.

Následne je tento vektor predaný funkcii `make_prediction`. Funkcia vykoná predikciu na vstupných dátach a vytvorí výstup obsahujúci výslednú klasifikáciu (buď „Irritant“ alebo „Non-Irritant“), dôveru predikcie (určenú ako heuristický „Trust Score“), cestu k obrázku ROC krivky, vizualizáciu SHAP hodnôt a zoznam najdôležitejších hyperparametrov, ktoré boli použité pri trénovaní modelu.

Okrem toho sa v prípade úspešnej predikcie získa pomocou knižnice PubChemPy dodatočná informácia o molekule, ako je IUPAC názov, molekulový vzorec a prípadne CAS číslo.

Súčasťou backendu je aj generovanie obrázkov molekulovej štruktúry. Tento krok zabezpečuje funkcia `get_molecule_img`, ktorá na základe SMILES reťazca vygeneruje 2D vizualizáciu molekuly, uloží ju do priečinka `static` a vráti cestu k obrázku, ktorá sa následne použije vo fronte.

Výsledná odpoveď je vrátená ako JSON objekt, ktorý obsahuje všetky údaje po-

trebné pre zobrazenie na stránke – výsledok klasifikácie, vizualizácie, chemické informácie aj modelové parametre.

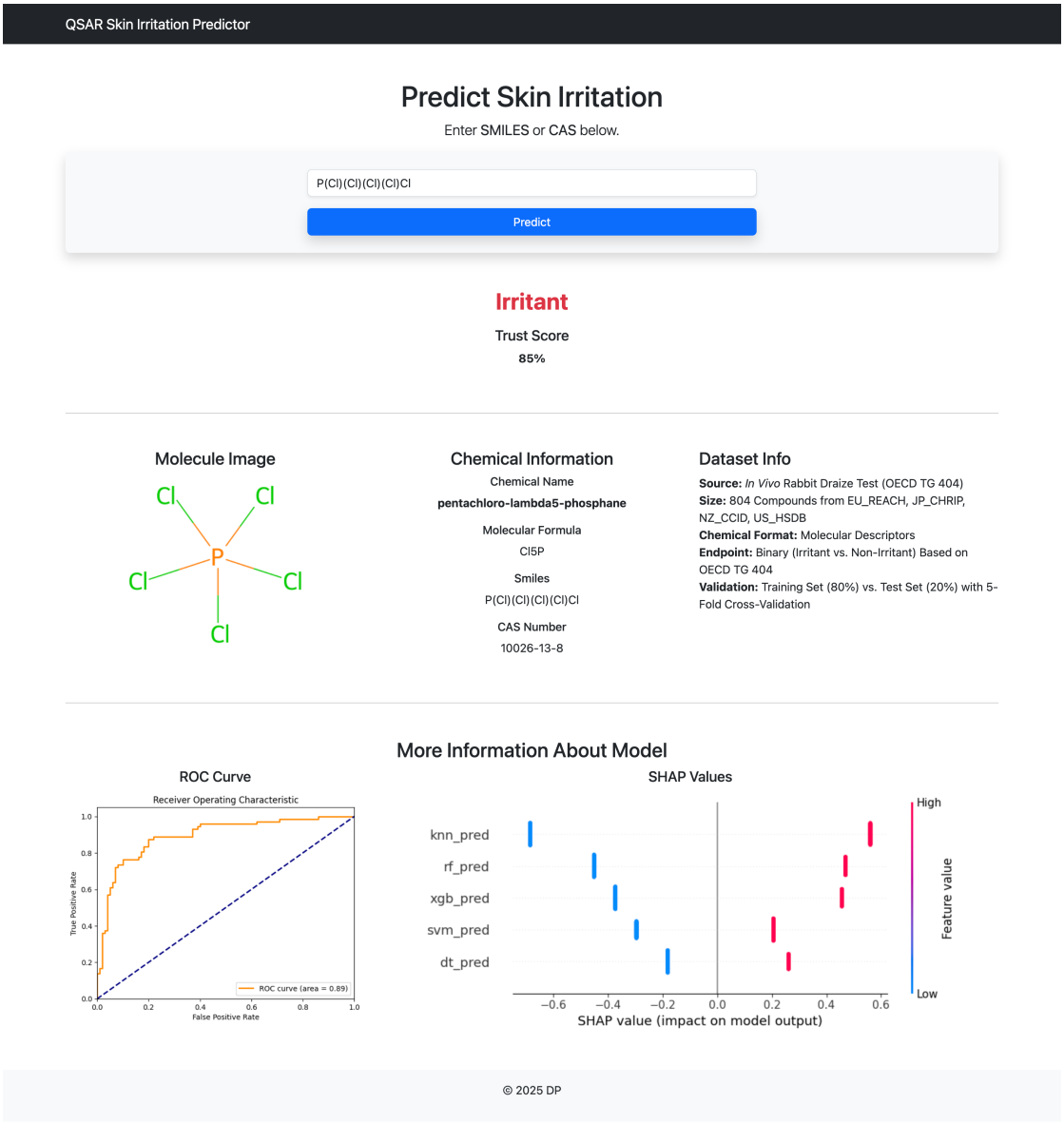
V prípade chybného vstupu alebo neúspešnej konverzie reťazca na molekulu je vyvolaná výnimka `ValueError` a používateľ dostane spätnú väzbu vo forme chybovej správy. .

3.9.2 Frontendová časť aplikácie

Frontendová časť aplikácie je vytvorená pomocou HTML šablóny doplnenej o štýly z frameworku Bootstrap a interaktívne prvky v JavaScripte. Hlavnou stránkou je šablóna `home.html`, ktorá slúži ako vstupné používateľské rozhranie pre zadanie vstupných údajov a zobrazenie výsledkov predikcie.

Rozhranie obsahuje textové pole pre zadanie vstupu, rozbaľovacie menu pre výber predikčného modelu a tlačidlo na spustenie predikcie. Po stlačení tlačidla sa prostredníctvom JavaScriptovej funkcie spustí asynchrónna AJAX požiadavka, ktorá odošle údaje na endpoint `/predict/` spolu s CSRF tokenom potrebným pre zabezpečenie Django aplikácií. Počas spracovania sa používateľovi zobrazí spinner s textom „Processing your input...“, čím sa zabezpečí spätná väzba o prebiehajúcej operácii.

Po prijatí odpovede zo servera je HTML stránka dynamicky aktualizovaná. Na základe obsahu odpovede sa zobrazí výsledok klasifikácie vo farebnom texte – červeným písmom, ak ide o iritujúcu látku, alebo zeleným, ak ide o neiritant.



Obr. 3.10: Obrazovka predikie

Zároveň sa zobrazí obrázok molekuly, výstupné hodnoty ako chemický názov, molekulový vzorec, SMILES a CAS číslo, a taktiež obrázky ROC krivky a SHAP vizualizácie, ktoré boli predgenerované pre každý model.

Celé rozhranie je navrhnuté tak, aby bolo responzívne a zrozumiteľné aj pre pou-

žívateľov bez technického zázemia. V prípade, že používateľ zadá neplatný vstup alebo dôjde k chybe počas predikcie, zobrazí sa chybové hlásenie vo forme modálneho okna alebo alertu, čím sa zabezpečí použiteľnosť systému aj pri výnimočných situáciách. Všetka logika sa vykonáva bez opätovného načítania stránky, čo výrazne zvyšuje komfort pri práci s aplikáciou.

3.9.3 Interakcia medzi komponentmi

Interakcia medzi jednotlivými komponentmi systému – teda medzi frontendovým rozhraním, backendovým serverom – prebieha na princípe klient-server komunikácie prostredníctvom HTTP protokolu. Používateľ začína interakciu zadaním vstupu. Po kliknutí na tlačidlo „Predict“ sa pomocou JavaScriptovej funkcie odošle asynchrónna POST požiadavka na backendový endpoint `/predict/`, pričom súčasťou požiadavky sú oba vstupy – chemická štruktúra a názov modelu.

Django backend túto požiadavku prijme a vstup spracuje prostredníctvom funkcie

Zhodnotenie

Diplomová práca sa zaoberala vývojom predikčného modelu na stanovenie dermatotoxicity chemických látok s cieľom poskytnúť efektívnu a etickú alternatívu k tradičným *in vivo* testom. Vzhľadom na rastúce požiadavky na ochranu verejného zdravia a zakázanie testovania na zvieratách je QSAR prístup čoraz viac využívaný.

V rámci práce bolo navrhnuté spracovanie dát, výpočet molekulárnych deskriptorov a výber relevantných vlastností pomocou SHAP, korelácie a LASSO. Následne boli implementované viaceré modely strojového učenia – Random Forest, Decision Tree, XGBoost, KNN a SVM – ktoré boli porovnané na základe klasifikačných metrík. Najlepšie výsledky dosiahol model XGBoost, ktorý sa ukázal ako najpresnejší.

Pre dosiahnutie stabilnejších výsledkov boli použité viaceré metódy, a to spojenie modelov pomocou meta-modelu logistickej regresie a vytvorenie finálnej predikcie hlasovaním. Oba prístupy ukázali, že dosahujú lepšie výsledky ako samostatné modely.

Súčasťou riešenia bola aj webová aplikácia, ktorá umožňuje používateľom nahrávať chemické zlúčeniny vo formáte SMILES a získať okamžitú predikciu dermatotoxicity. Predikcie vykonával model, ktorý vznikol spojením modelov pomocou meta-

modelu logistickej regresie, keďže dosahoval najlepšie výsledky. Aplikácia zároveň ponúka interpretovateľné výstupy prostredníctvom SHAP grafov na vysvetlenie výsledkov modelov.

Výsledky práce potvrdzujú, že spojením QSAR a strojového učenia je možné efektívne predikovať dermatotoxicitu chemických látok. Navrhnutý systém môže slúžiť ako nástroj pre predbežné hodnotenie bezpečnosti látok, čím prispieva k ochrane zdravia a znižovaniu závislosti na *in vivo* metódach testovania.

3.10 Možné vylepšenia a budúci vývoj

Je je niekoľko možností, ktoré by mohli pomôcť k rozšíreniu a zlepšeniu vyvinutého riešenia pri budúcom vývoji. Prvým krokom by mohlo byť rozšírenie datasetu, čo by znamenalo zvýšenie množstva a diverzity dát. Tento krok by mal za následok zlepšenie generalizácie modelov, čím by sa zvýšila ich schopnosť efektívne pracovať aj s menej častými chemickými štruktúrami. Ďalším možným krokom je podpora multi-label klasifikácie, ktorá by umožnila súčasne predikovať viacero druhov toxicít, ako je napríklad iritácia, senzibilizácia alebo karcinogenita. Tento prístup by zvýšil univerzálnosť modelu a umožnil širšie využitie v rôznych oblastiach výskumu a praxe. Nakoniec, zavedenie spôsobu úpravy predikcií, kde by užívateľ mohol upraviť predikciu na základe skutočného testovania, čím by sa model v priebehu času zlepšoval a prispôsoboval novým poznatkom. Tieto kroky by mohli prispieť k zvýšeniu presnosti a efektivity vyvinutého riešenia v praxi.

Literatúra

[1] Namita Agarwal a Saikat Das. "Interpretable machine learning tools: A survey". In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2020, s. 1528–1534.

[2] Plain English AI. *Introduction to K-Nearest Neighbors (KNN) Algorithm*. Accessed: 2025-01-02. n.d. URL: https://ai.plainenglish.io/introduction-to-k-nearest-neighbors-knn-algorithm-e8617a448fa8.

[3] Anastasia Anceschi et al. "Keratose Self-Cross-Linked Wound Dressing for Iron Sequestration in Chronic Wounds". In: *ACS omega* 8.33 (2023), s. 30118–30128.

[4] Jochen Kühnl Anke Wilm a Johannes Kirchmair. "Computational approaches for skin sensitization prediction". In: *Critical Reviews in Toxicology* 48.9 (2018). PMID: 30488745, s. 738–760. DOI: 10.1080/10408444.2018.1528207. eprint: https://doi.org/10.1080/10408444.2018.1528207. URL: https://doi.org/10.1080/10408444.2018.1528207.

[5] Shekhar Banerjee. *Deciphering Decision Trees: The Art of Informed Choices*. Accessed: 2025-01-02. n.d. URL: https://shekhar-banerjee96.medium.com/deciphering-decision-trees-the-art-of-informed-choices-e95fd85ea1f5.

[6] ChemIntelligence. *Machine Learning and Descriptors of Molecules*. Accessed: 2025-01-03. 2025. URL: https://chemintelligence.com/blog/machine-learning-descriptors-molecules (cit. 03. 01. 2025).

[7] Tianqi Chen a Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, aug. 2016. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[8] Artem Cherkasov et al. "QSAR Modeling: Where Have You Been? Where Are You Going To?" In: *Journal of Medicinal Chemistry* 57.12 (2014). PMID: 24351051, s. 4977–5010. DOI: 10.1021/jm4004285. URL: https://doi.org/10.1021/jm4004285.

[9] Caroline Cooper a Rupert Purchase. "Representation of Organic Compounds: Molecular Formulae, CAS Registry Numbers and Linear Notations". In: *Organic Chemist's Desk Reference*. Taylor a Francis, 2017. Kap. 15. DOI: 10.4324/9781315120768-15. URL: https://www.taylorfrancis.com/chapters/edit/10.4324/9781315120768-15/representation-organic-compounds-molecular-formulae-cas-registry-numbers-linear-notations-caroline-cooper-rupert-purchase.

[10] Danishuddin a Asad U. Khan. "Descriptors and their selection methods in QSAR analysis: paradigm for drug design". In: *Drug Discovery Today* 21.8 (2016), s. 1291–1302. ISSN: 1359-6446. DOI: https://doi.org/10.1016/j.drudis.2016.06.013. URL: https://www.sciencedirect.com/science/article/pii/S1359644616302318.

[11] Rifkat Davronov a Samariddin Kushmuratov. "Comparative analysis of QSAR feature selection methods". In: *AIP Conference Proceedings* 3004.1 (mar. 2024), s. 050002. ISSN: 0094-243X. DOI: 10.1063/5.0199872. eprint: https:

//pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0199872/19722248/050002_1_5.0199872.pdf. URL: https://doi.org/10.1063/5.0199872.

[12] Yi Ding et al. "Molecular fingerprint-based machine learning assisted QSAR model development for prediction of ionic liquid properties". In: *Journal of Molecular Liquids* 326 (2021), s. 115212. ISSN: 0167-7322. DOI: https://doi.org/10.1016/j.molliq.2020.115212. URL: https://www.sciencedirect.com/science/article/pii/S0167732220374547.

[13] Martin Eklund et al. "Choosing Feature Selection and Learning Algorithms in QSAR". In: *Journal of Chemical Information and Modeling* 54.3 (2014). PMID: 24460242, s. 837–843. DOI: 10.1021/ci400573c. eprint: https://doi.org/10.1021/ci400573c. URL: https://doi.org/10.1021/ci400573c.

[14] Mohammad Goodarzi, Bieke Dejaegher a Yvan Vander Heyden. "Feature selection methods in QSAR studies". English. In: *Jouran of AOAC International* 95.3 (2012), s. 636–651. ISSN: 1060-3271.

[15] Rajarshi Guha. "On the interpretation and interpretability of quantitative structure–activity relationship models". In: *Journal of computer-aided molecular design* 22 (2008), s. 857–871.

[16] Mohammad Hossin a Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations". In: *International journal of data mining & knowledge management process* 5.2 (2015), s. 1.

[17] Gabriel Idakwo et al. "A Review of Feature Reduction Methods for QSAR-Based Toxicity Prediction". In: *Advances in Computational Toxicology: Methodologies and Applications in Regulatory Science*. Ed. Huixiao Hong. Cham: Springer International Publishing, 2019, s. 119–139. ISBN: 978-3-030-16443-

0. DOI: 10.1007/978-3-030-16443-0_7. URL: https://doi.org/10.1007/978-3-030-16443-0_7.

[18] Vikramaditya Jakkula. "Tutorial on support vector machine (svm)". In: *School of EECS, Washington State University* 37.2.5 (2006), s. 3.

[19] A. Jović, K. Brkić a N. Bogunović. "A review of feature selection methods with applications". In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015, s. 1200–1205. DOI: 10.1109/MIPRO.2015.7160458.

[20] Shilpa Kamishetty. *XGBoost*. Accessed: 2025-01-02. n.d. URL: https://medium.com/@shilpakamishetty/xgboost-350530593cb5.

[21] Bin Li et al. "Deep eutectic solvent self-assembled reverse nanomicelles for transdermal delivery of sparingly soluble drugs". In: *Journal of Nanobiotechnology* 22.1 (2024), s. 272.

[22] Shuxia Lu a Mi Zhou. "Log-loss SVM Classification for Imbalanced Data". In: *Int. J. Recent Eng. Sci* 4.2 (2017), s. 7–10.

[23] Wilson E. Marcílio a Danilo M. Eler. "From explanations to feature selection: assessing SHAP values as feature selection mechanism". In: *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2020, s. 340–347. DOI: 10.1109/SIBGRAPI51738.2020.00053.

[24] Mariia Matveieva a Pavel Polishchuk. "Benchmarks for interpretation of QSAR models". In: *Journal of cheminformatics* 13.1 (2021), s. 41.

[25] Hu Mei et al. "Support vector machine applied in QSAR modelling". In: *Chinese Science Bulletin* 50 (2005), s. 2291–2296.

[26] Aiman Moldagulova a Rosnafisah Bte. Sulaiman. "Using KNN algorithm for classification of textual documents". In: *2017 8th International Conference on Information Technology (ICIT)*. 2017, s. 665–671. DOI: 10.1109/ICITECH.2017.8079924.

[27] Eugene N. Muratov et al. "QSAR without borders". In: *Chem. Soc. Rev.* 49 (11 2020), s. 3525–3564. DOI: 10.1039/D0CS00098A. URL: http://dx.doi.org/10.1039/D0CS00098A.

[28] OECD. *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models*. 2014, s. 154. DOI: https://doi.org/https://doi.org/10.1787/9789264085442-en. URL: https://www.oecd-ilibrary.org/content/publication/9789264085442-en.

[29] J. Padarian, A. B. McBratney a B. Minasny. "Game theory interpretation of digital soil mapping convolutional neural networks". In: *SOIL* 6.2 (2020), s. 389–397. DOI: 10.5194/soil-6-389-2020. URL: https://soil.copernicus.org/articles/6/389/2020/.

[30] Amit Pandey a Achin Jain. "Comparative analysis of KNN algorithm using various normalization techniques". In: *International Journal of Computer Network and Information Security* 11.11 (2017), s. 36.

[31] Aakash Parmar, Rakesh Katariya a Vatsal Patel. "A review on random forest: An ensemble classifier". In: *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*. Springer. 2019, s. 758–763.

[32] Arti Patle a Deepak Singh Chouhan. "SVM kernel functions for classification". In: *2013 International conference on advances in technology and engineering (ICATE)*. IEEE. 2013, s. 1–9.

[33] Gabriel A. Pinheiro et al. "Machine Learning Prediction of Nine Molecular Properties Based on the SMILES Representation of the QM9 Quantum-Chemistry Dataset". In: *The Journal of Physical Chemistry A* 124.47 (2020). PMID: 33174750, s. 9854–9866. DOI: 10.1021/acs.jpca.0c05969. eprint: https://doi.org/10.1021/acs.jpca.0c05969. URL: https://doi.org/10.1021/acs.jpca.0c05969.

[34] Steven J Rigatti. "Random forest". In: *Journal of Insurance Medicine* 47.1 (2017), s. 31–39.

[35] Raquel Rodríguez-Pérez a Jürgen Bajorath. "Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions". In: *Journal of computer-aided molecular design* 34.10 (2020), s. 1013–1026.

[36] Mohsen Shahlaei. "Descriptor Selection Methods in Quantitative Structure–Activity Relationship Studies: A Review Study". In: *Chemical Reviews* 113.10 (2013). PMID: 23822589, s. 8093–8103. DOI: 10.1021/cr3004339. eprint: https://doi.org/10.1021/cr3004339. URL: https://doi.org/10.1021/cr3004339.

[37] *SHAP*. URL: https://kirenz.github.io/mlops/shap.html (cit. 06. 02. 2024).

[38] Tarapong Srisongkram et al. "Stacked ensemble learning on HaCaT cytotoxicity for skin irritation prediction: A case study on dipterocarpol". In: *Food and Chemical Toxicology* 181 (2023), s. 114115.

[39] Vladimir Svetnik et al. "Random forest: a classification and regression tool for compound classification and QSAR modeling". In: *Journal of chemical information and computer sciences* 43.6 (2003), s. 1947–1958.

[40] Yaseen Taha a Abdulhamit Subhi Abdulazeez. "Classification Based on Decision Tree Algorithm for Machine Learning". In: *Journal of Applied Science and Technology Trends* 2.01 (2021), s. 20–28. DOI: 10.38094/jastt20165. URL: https://www.jastt.org/index.php/jasttpath/article/view/65.

[41] Alexander Tropsha. "Best Practices for QSAR Model Development, Validation, and Exploitation". In: *Molecular Informatics* 29.6-7 (2010), s. 476–488. DOI: https://doi.org/10.1002/minf.201000061. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/minf.201000061. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201000061.

[42] U.S. Food and Drug Administration. *New developments in regulatory QSAR modeling: a new QSAR model for predicting blood brain barrier permeability.* Accessed: 2025-01-02. 2023. URL: https://www.fda.gov/drugs/regulatory-science-action/new-developments-regulatory-qsar-modeling-new-qsar-model-predicting-blood-brain-barrier-permeability.

[43] Vitalflux. *Classification Model – SVM Classifier Python Example.* Accessed: 2025-01-02. n.d. URL: https://vitalflux.com/classification-model-svm-classifier-python-example/.

[44] Ž Vujović et al. "Classification model evaluation metrics". In: *International Journal of Advanced Computer Science and Applications* 12.6 (2021), s. 599–606.

[45] Apilak Worachartcheewan et al. "Large-scale QSAR study of aromatase inhibitors using SMILES-based descriptors". In: *Chemometrics and Intelligent Laboratory Systems* 138 (2014), s. 120–126. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2014.07.017. URL: https://www.sciencedirect.com/science/article/pii/S0169743914001646.

[46] Jiaju Wu et al. "Prediction and Screening Model for Products Based on Fusion Regression and XGBoost Classification". In: *Computational Intelligence and Neuroscience* 2022 (2022), s. 4987639. DOI: 10.1155/2022/4987639.

[47] Shichao Zhang. "Challenges in KNN Classification". In: *IEEE Transactions on Knowledge and Data Engineering* 34.10 (2022), s. 4663–4675. DOI: 10.1109/TKDE.2021.3049250.

[48] Shichao Zhang et al. "Learning k for knn classification". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 8.3 (2017), s. 1–19.