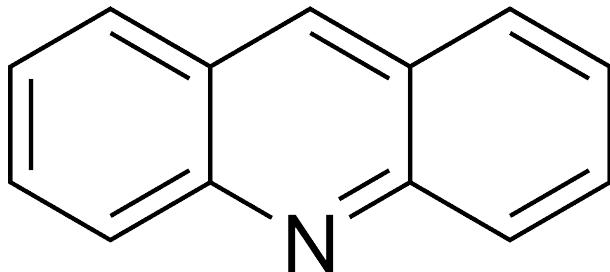


SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

TÍMOVÝ PROJEKT



FOTOTOX

Vedúca tímu:

Ing. Marta Šoltésová Prnová, PhD.

Členovia tímu:

Bc. Matej Halinkovič

Bc. Jakub Knánik

Bc. Jakub Maruniak

Bc. Kateřina Mušková

Bc. Patrik Pavlačka

Bc. Tibor Sloboda

Kontakt:

tim20.tp2021@gmail.com

team20-21.studenti.fit.stuba.sk

Contents

I Inžinierske dielo	4
1 Úvod	5
2 Globálne ciele projektu	6
2.1 Zimný semester	6
2.2 Letný semester	6
3 Celkový pohľad na systém	8
4 Moduly systému	9
4.1 Data pipe	9
4.2 Trénovanie modelov	11
4.3 Webová aplikácia	13
4.3.1 ML Ops	15
4.3.2 Obrazovky webovej aplikácie	17
4.3.3 Testovanie webové aplikace	22
5 Používateľská príručka	23
5.1 Domovská obrazovka	23
5.2 Registrácia/prihlásenie	24
5.3 Predikcia	24
5.4 Odovzdávanie nových dát	25
5.5 Validovanie dát	26
II Riadenie projektu	28
1 Úvod	29
2 Role členov tímu a podieľ práce	30
2.1 Role členov tímu	30
2.2 Podieľ práce	31
2.2.1 Zimný semester	31
2.2.2 Letný semester	32

CONTENTS	2
3 Aplikácie manažmentov	37
3.1 Manažment kvality	37
3.2 Manažment testovania	37
3.3 Manažment verziovania projektu	37
3.4 Manažment správy úloh	38
3.5 Manažment dátovej vedy	39
3.6 Manažment komunikácie	39
4 Globálna retrospektíva	40
4.1 Zimný semester	40
4.2 Letný semester	40
5 Sumarizácie šprintov	41
5.1 Zimný semester	41
5.2 Letný semester	51
Prílohy	A-1
A Metodika zabezpečovania kvality kódu	A-1
B Metodika testovania	B-1
C Metodika verziovania kódu	C-1
D CICD Pipeline	D-1
E Metodika správy úloh	E-1
E.1 Workflow	E-2
E.2 Typy úloh	E-3
E.3 Role	E-4
E.4 Komunikácia	E-4
F Metodika dátovej vedy	F-1
G Používané chemické deskriptory	G-1
H Metodika komunikácie	H-1
I Metodika ukladania dokumentov a informácií	I-1

CONTENTS	3
J Export úloh	J-1
K Prihláška na TP Cup	K-1
L Motivačný dokument	L-1
M Protokol z testovania	N-1
N Generovaná technická dokumentácia	N-2

PART I:

INŽINIERSKE DIELO

1 Úvod

V tímovom projekte FOTOTOX sa venujeme predikcii fototoxicity chemickej látok prostredníctvom metódy *in silico*. Tento dokument slúži ako dokumentácia k danému projektu.

V tejto časti dokumentácie sa možno dočítať o cieľoch projektu, z akých častí vyvijaný systém pozostáva a z akých modulov sú dané časti systému vytvorené. Pozrieme sa na architektúru riešenia a funkcionality jednotlivých častí modulov.

2 Globálne ciele projektu

Hlavným cieľom projektu je schopnosť predikovať fototoxicitu chemickej látky. Súčasťou dosiahnutia tohto cieľa je využitie modelov kvantitatívnej závislosti aktivity na štruktúre (QSAR), ktorými možno získať vzťahy medzi chemickou štruktúrou a biologickou aktivitou v súbore dát. Zo zhromaždených dát a s pomocou natrénovaného modelu budeme následne vedieť predikovať fototoxicitu látky, pričom výpočet fototoxicity bude prístupný pre používateľa prostredníctvom webovej aplikácie.

2.1 Zimný semester

Hlavné ciele zimného semestra:

- Zadefinovanie si postupu práce a metodík.
- Získanie dostatočného množstva dát.
- Porozumenie problematike fototoxicity. Vykonanie EDA pre získané dátu.
- Identifikovanie a vypočítanie chemických deskriptorov, ktoré by mohli korelovať s fototoxicitou.
- Návrh celkovej architektúry.
- Návrh a implementácia modulu data pipe.
- Návrh a implementácia machine learning modelu.
- Návrh webovej aplikácie.
- Dodržiavanie agile postupov. Iteratívno-inkrementálny prístup k tvorbe softvéru.

2.2 Letný semester

Celkový pohľad po zimnom semestri:

- Zmeny v manažmente od začiatku letného semestra ovplyvnili aj tvorbu inžinierskeho diela - jednotyždňové šprinty, definition of done a story pointy. Tieto zmeny nám umožnili efektívnejšie pridávať potrebnú funkcia&onalitu a flexibilnejšie sa prispôsobiť novovzniknutým problémom.
- Po zimnom semestri bolo našou prioritou nasadenie prototypu aplikácie na virtuálny stroj.
- Identifikovali sme potrebu redizajnu webovej aplikácie, aby splňala naše požiadavky (napr. autentifikácia, pridávanie návrhov pre nové látky, updatovanie modelu a pod.)

Hlavné ciele letného semestra:

- Vylepšenie súčasných metód predikcie. Preskúmanie možných vylepšení využitím neruónových sietí (s možným významným vylepšením pomocou grafových neurónových sietí).
- Redizajn webovej aplikácie.
- Implementácia všetkých kľúčových funkcia&onalít webovej aplikácie.
- Nasadenie webovej aplikácie.
- Dodržiavanie agile postupov ako aj zmien v manažmente pre letný semester.
- Vykonanie potrebných aktivít pre úspešné absolvovanie IIT.SRC a TPCup.

3 Celkový pohľad na systém

Zo špecifikácie nášho projektu vyplynulo rozdelenie celkového systému na dve logické časti.

1. Vytvorenie modelu

Prvou časťou nášho projektu je vytváranie spôsobu ako predikovať fotoxicitu látky. Za týmto účelom používame prístupy strojového učenia, ktoré vyžadujú získavanie dát, predspracovanie dát, trénovanie modelov a ich validácia.

2. Vytvorenie web aplikácie

Druhou časťou nášho projektu je zdostupnenie nášho modelu pre verejnosc. Zvolili sme spôsob vývoja web aplikácie, ktorá nám pomôže dosiahnuť čo najširišie využitie.

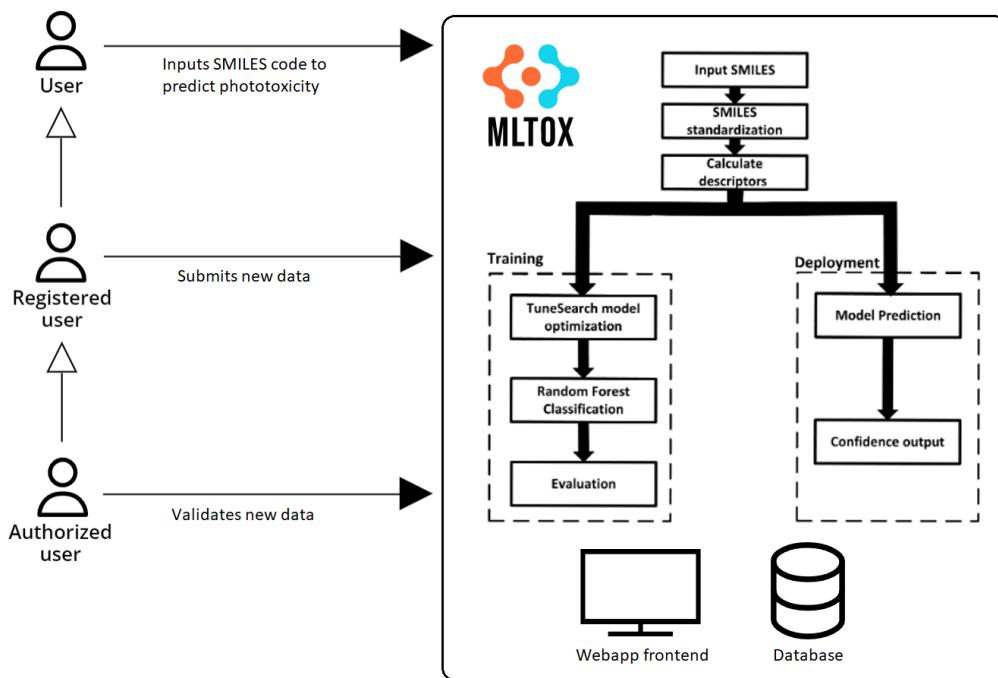


Figure 1: Prehľad webovej aplikácie MLTOX

4 Moduly systému

4.1 Data pipe

Získavanie dát pre našu problematiku je iteratívny proces. Overovanie fototoxicity pre látky je zdĺhavé a môžeme preto spracovať zväčša len pári látok naraz a požiadavky na predspracovanie deskriptorov sa vyvíjajú. Z týchto dôvodov bolo pre nás dôležité zostaviť proces, ktorý nám umožnil automatizovať značnú čas práce a sústredíť sa tak na vývoj čo najlepšieho modelu.

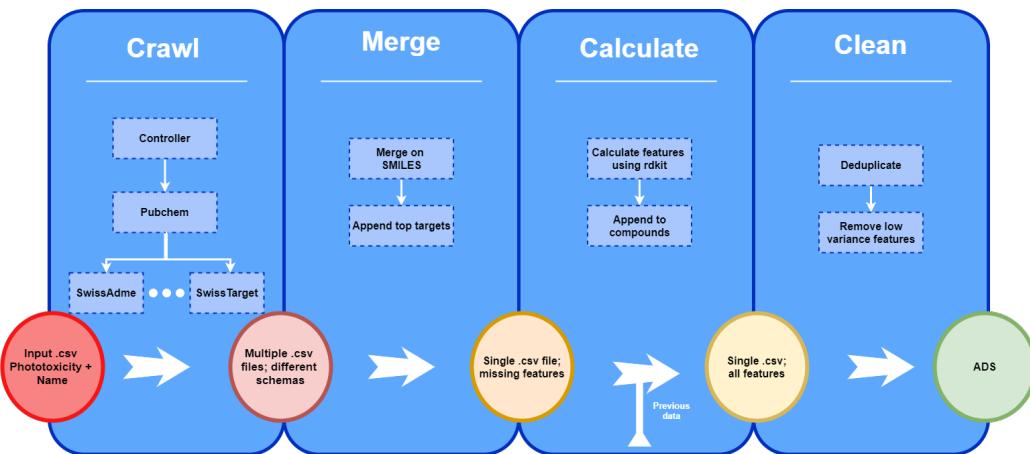


Figure 2: Spôsob extrakcie a predspracovania dát pre modely

1. Crawl

Názvy látok a informácie o ich fototoxicite získavame zväčša manuálne z vedeckých publikácií. Tieto publikácie však neobsahujú hodnoty deskriptorov, ktoré potrebujeme na predikciu. Preto sme zostavili niekoľko crawlerov, ktoré zoberú zoznam látok, ktorý im zadáme a hodnoty deskriptorov vedia extrahovať z chemických databáz.

2. Merge

Výstupy z našich crawlerov potrebujeme zjednotiť do jedného súboru pre jednoduchšie spracovanie v neskôrších krokoch. Databázy pracujú na princípe SMILES kódov, ktoré používame ako kľúč pre zlučovanie.

3. Calculate

Na základe poznatkov získaných z prednášok o fototoxicite sme identifikovali niekoľko užitočných deskriptorov, ktoré sa nezvyknú nachádzať v databázach. Pre ich vypočítanie používame knižnicu Rdkit. V tomto kroku pripájame informácie o nových látkach k už existujúcemu zoznamu látok z predchádzajúcich iterácií. Tento krok je potrebný pre zabezpečenie niektorých častí funkcionality čistenia dát.

4. Clean

V poslednom kroku prípravy dát pre experimenty ich čistíme. Pre zaručenie stability našich modelov odstraňujeme premenné s nízkou variáciou, premenné ktoré sú vysoko korelované s inými a zabezpečujeme, že náš dataset neobsahuje duplikované látky.

Pre overenie správnosti funkcionality tejto pipeline, sme vytvorili aj niekoľko testov podľa metodiky testovania, bližšie popísanej v Prílohe B.

4.2 Trénovanie modelov

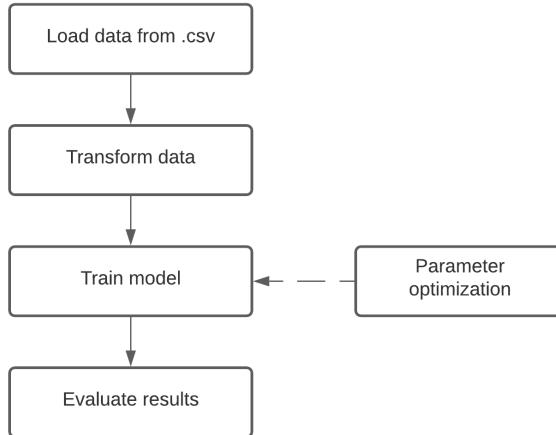


Figure 3: Štruktúra experimentov

Základ nášho systému tvorí model, ktorý dokáže predikovať fototoxicitu látky. Zostavili sme experimentálny proces, ktorý nám dovolí jednoducho natreňovať optimálny model pre zadaný dataset stlačením jedného tlačidla.

Proces je implementovaný pomocou Jupyter Notebook a pre optimalizáciu modelu je využitý rámec TuneSearch. Tento proces nám umožňuje automaticky nájsť najlepšiu kombináciu modelu a jeho parametrov pre daný problém.

Úspešnosť modelov je vyhodnocovaná na základe nasledujúcich metrík:

Metrika	Výpočet
Správnosť	$\frac{TP+TN}{TP+TN+FP+FN}$
Presnosť	$\frac{TP}{TP+FP}$
Úplnosť	$\frac{TP}{TP+FN}$
F1-skóre	$2 * \frac{\text{presnosť} * \text{úplnosť}}{\text{presnosť} + \text{úplnosť}}$

Table 1: TP - skutočná pozitivita (fototoxická látka predikovaná ako fototoxická), FP - falošná pozitivita (nefototoxická látka predikovaná ako fototoxická), TN - skutočná negativita (nefototoxická látka predikovaná ako nefototoxická), FN - falošná negativita (nefototoxická látka predikovaná ako fototoxická)

Modely vyhodnocujeme aj podľa systému SHAP, ktorý poskytuje informácie o dôležitosti jednotlivých premenných pre rozhodnutia modelu. Tento prístup nám umožňuje rozhodnúť či rozhodnutia dávajú zmysel z chemického hľadiska a pomáha nám určovať spoľahlivosť modelu. Presný popis použitých deskriptorov a ich dôležitosť pre rozhodnutia modelu je možné vidieť v Prílohe G.

Aktuálne najlepší model je postavený na optimalizovanej RandomForest architekúre a dosahuje nasledovné výsledky:

Metrika	Hodnota
Správnosť	78.58%
Presnosť	89.47%
Úplnosť	80.95%
F1-skóre	85.00%

4.3 Webová aplikácia

Backend a frontend webovej aplikácie sme implementovali vo frameworku Django. Pre frontend webovej aplikácie sme taktiež implementovali skripty v jazyku JavaScript pre zrýchlenie a zlepšenie použiteľnosti našej webovej aplikácie. Aplikácia pozostáva zo 4 hlavných modulov:

- Modul autentifikácia implementuje backend aplikácie pre prihlásenie a registráciu používateľa a taktiež obsahuje implementáciu pre zmenu hesla a reset hesla, čo umožňuje používateľovi nastaviť si nové heslo, ak zabudol staré heslo.
- Modul compound analyzer implementuje biznis logiku aplikácie pre predikovanie, pre zobrazenie výsledkov predikcií, pre pridávanie nových experimentálnych dát, pre korekciu predikcie a pre validáciu prijatých experimentálnych dát.
- Modul locale obsahuje preklady textov, ktoré sa zobrazujú používateľom do slovenčiny a angličtiny. Používateľ si môže nastaviť jazyk aplikácie.
- Modul MLOps je podrobnejšie popísaný v sekcií 4.3.1.

Pre predikovanie fototoxicity stačí zadať do aplikácie len SMILES kód. Pri predikovaní fototoxicity najprv skontrolujeme, či nemáme danú látku uloženú v databázovej tabuľke Molecule. Ak danú látku máme uloženú, tak len vrátime výsledok z databázy. Ak danú látku nemáme uloženú v databázovej tabuľke Molecule, tak zo zadaneho SMILES kódu vypočítame chemické deskriptory a tie potom pošleme modelu pre predikciu fototoxicity. Logický dátový model aplikácie môžeme vidieť na obrázku 4. Predikovať fototoxicitu je umožnené všetkým používateľom aj bez registrácie. Ostatné funkcionality aplikácie sú dostupné len pre registrovaných používateľov, pričom validácia prijatých experimentálnych dát je dostupná len pre chemikov, ktorí s nami spolupracujú pri vývoji našej aplikácie. Momentálne je umožnené registrovať sa len používateľom, ktorí pôsobia na akademických inštitúciach a vlastnia e-mailové adresy s doménou danej akademickej inštitúcie.

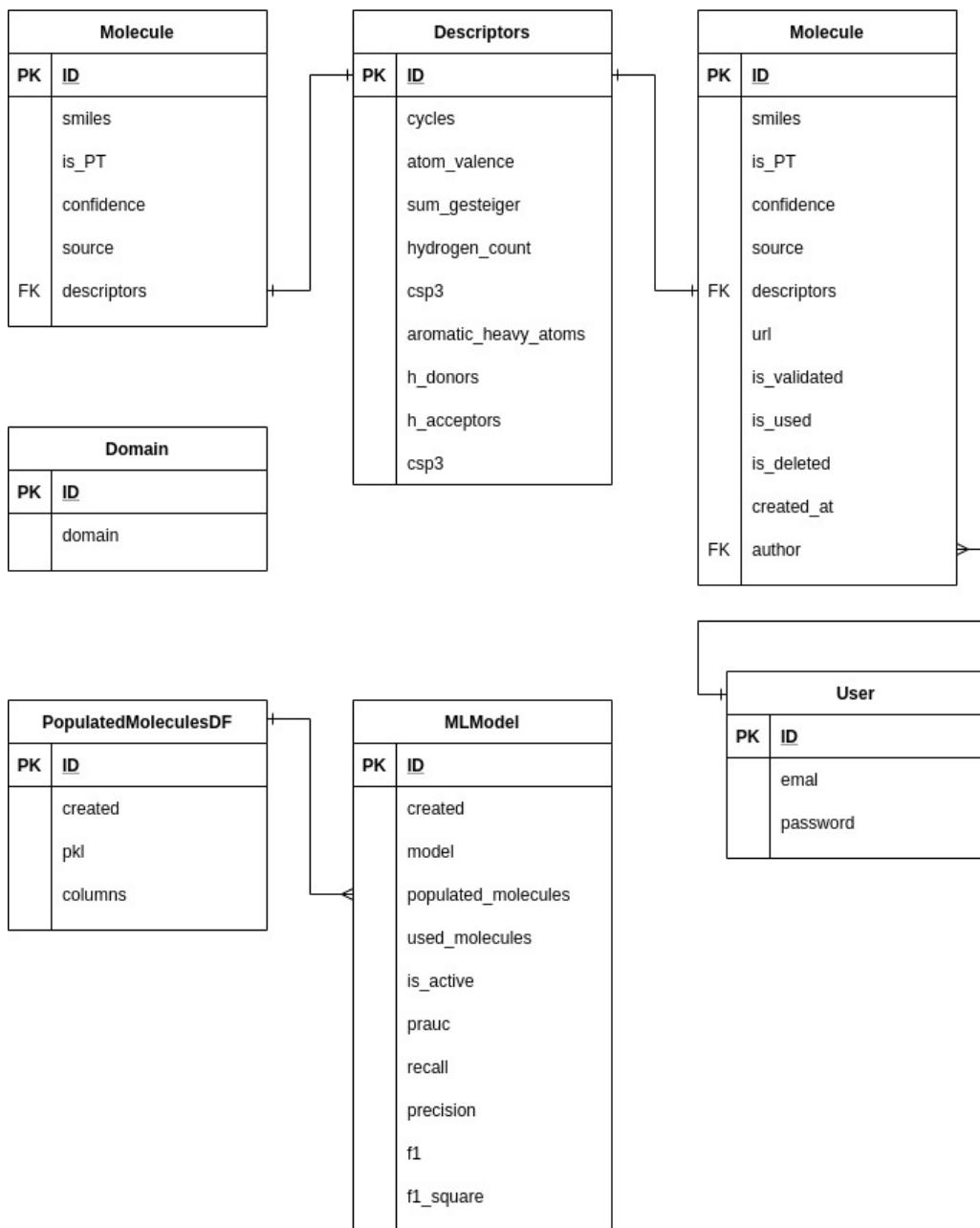


Figure 4: Logický dátový model aplikácie

4.3.1 MLOps

Web-aplikáce je strukturovaná tak, aby sa automaticky nasazovali modely v momente, kdy data o validovaných návrhach prekročí určité metriky.

Pokud k tomuto stuvu doje, natrénuje se několik modelů, jejichž vstupem jsou všechny molekuly v databázi a podmnožina nových návrhů. Každý model je optimalizovaný pomocí HyperTune nástroje.

Po natrénování jsou tyto modely porovnány mezi sebou, nejlepší pak s existujícím modelem. Pokud jsou metriky nového modelu lepší, tak nahrazuje model stávající.

Každý použitý model zůstává v databázi. Spolu s modelem samotným jsou uloženy i data, jako jeho metriky, vstupní data a typy použitých deskriptorů.

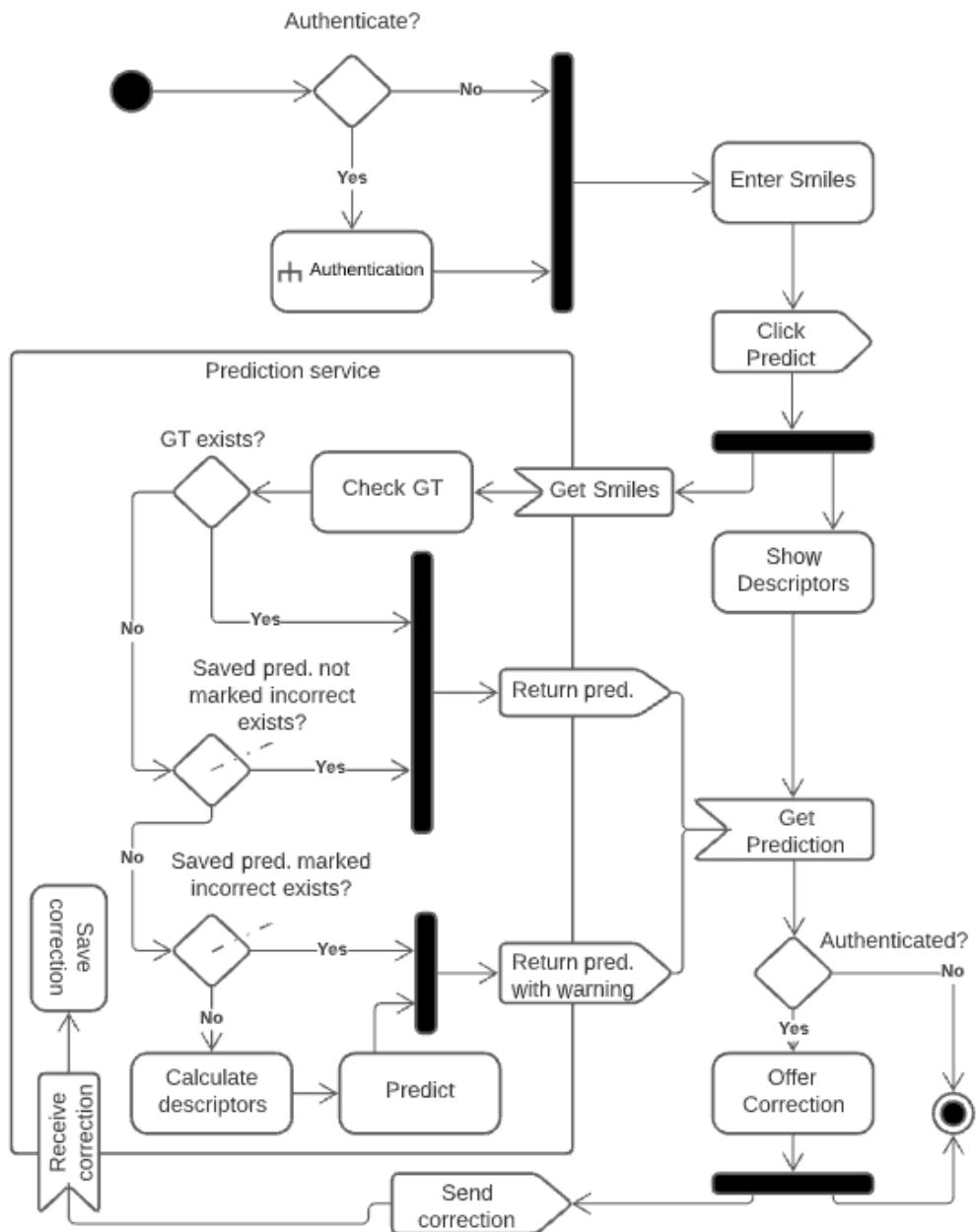


Figure 5: Diagram aktivít

4.3.2 Obrazovky webovej aplikácie

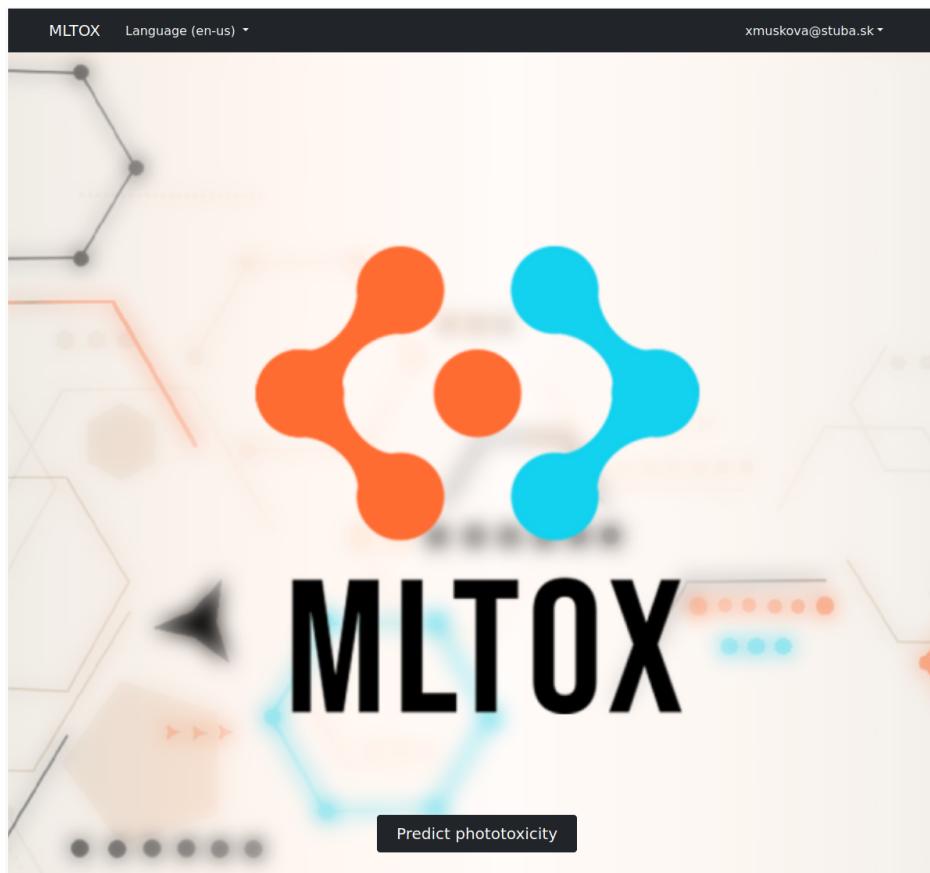


Figure 6: Landing page stránky.

The screenshot shows the 'Sign up' form on the MLTOX website. At the top, there is a navigation bar with 'MLTOX' and 'Language (en-us)'. On the right side of the bar are 'Log in' and 'Sign up' buttons. The main form has a title 'Sign up'. It contains three input fields: 'Email' (with value 'katerina.muskova@email.cz'), 'Password' (with placeholder 'Password'), and 'Password repeat' (with placeholder 'Confirm Password'). Below these fields is a yellow 'Sign Up' button. At the bottom of the form, there is a link 'Already have an account? [Log in](#)'. A red error message is displayed above the input fields: 'Email domain is not authorized for sign-up! To authorize this email domain send an email to tim20.tp2021@gmail.com.' There is a small red 'X' icon next to the message.

Figure 7: Overovanie identity používateľa, pri predikcii nieje potrebná ak nechce korigovať výsledky.

The screenshot shows the 'Predict phototoxicity' page on the MLTOX website. At the top, there is a navigation bar with 'MLTOX' and 'Language (en-us)'. On the right side of the bar are 'Log in' and 'Sign up' buttons. Below the bar, there is a breadcrumb trail: 'Home > Predict'. The main form has a title 'Predict phototoxicity'. It features a single input field labeled 'SMILES:' containing the chemical structure 'CC1=CC(=C(C=C1)O)N2N=C3C=CC=CC3=N2'. To the right of this input field is a red 'X' icon. Below the input field are two buttons: 'Submit' and 'Add Row'.

Figure 8: Vkladanie SMILES kódov pre predikciu.

Prediction results

1.SMILES: CCO

Molecule model:	Descriptors:	Result:
	Cycles: 0.0 Atom valence: 16.0 Positive sum: 0.43850220093603226 Hydrogen count: 6.0 CSP3: 1.0 Aromatic heavy atoms: 0.0 Hydrogen donors: 1.0 Hydrogen acceptors: 1.0	NPT 100% Confidence Correct the prediction

[Back to predicting](#)

Figure 9: Viditeľný model molekuly, pári základných deskriptorov a výsledok predikcie. Možnosť opravy predikcie ak je používateľ overený.

Prediction results

1.SMILES: CCO

Molecule model:	Descriptors:	Result:
	Cycles: 0.0 Atom valence: 16.0 Positive sum: 0.43850220093603226 Hydrogen count: 6.0 CSP3: 1.0 Aromatic heavy atoms: 0.0 Hydrogen donors: 1.0 Hydrogen acceptors: 1.0	Test used: 3T3 Correct result: NPT Enter a URL: https://example.com Submit Cancel

[Back to predicting](#)

Figure 10: Používateľ vie zadať typ testu, výsledok a pridať experimentálne dátá na opravu.

The screenshot shows the 'Prediction results' page of the MLTOX web application. At the top, there are navigation links: 'MLTOX Language (en-us)', 'xmuskova@stuba.sk', 'Home > Predict > Results', and a back link 'Back to predicting'. The main title is 'Prediction results'. Below it, a section titled '1.SMILES: CCO' displays a chemical structure of a hydroxyl group (OCC) and its properties:

Molecule model:	Descriptors:	Result:
	Cycles: 0.0	
	Atom valence: 16.0	
<chem>OCC</chem>	Positive sum: 0.43850220093603226	
	Hydrogen count: 6.0	
	CSP3: 1.0	
	Aromatic heavy atoms: 0.0	
	Hydrogen donors: 1.0	
	Hydrogen acceptors: 1.0	

To the right of the table, a message says 'Thank you for your submission!'. At the bottom, there is a 'Back to predicting' button.

Figure 11: Potvrdenie korekcie predikcie.

The screenshot shows the 'Submit experimental data' page of the MLTOX web application. At the top, there are navigation links: 'MLTOX Language (en-us)', 'xmuskova@stuba.sk', 'Home > Submit data', and a back link 'Back to predicting'. The main title is 'Submit experimental data'. The form contains two rows for SMILES input:

SMILES:	Test Type:	Result:
<chem>CCO</chem>	3T3	NPT X

SMILES:	Test Type:	Result:
<chem>CO</chem>	3D	NPT X

Below the rows, there is a URL input field with the value 'https://example.com' and a note 'Enter a URL: ?'. At the bottom, there are 'Submit' and 'Add Row' buttons.

Figure 12: Používateľ vie zadať ľubovoľný počet riadkov výsledkov, a pridať odkaz na existujúci publikáciu.

MLTOX Language (en-us) xmuskova@stuba.sk

[Home](#) > Submit data

Submit experimental data

SMILES:	Test Type:	Result:
CCO	3T3	NPT X
SMILES:	Test Type:	Result:
CO	3D	NPT X

Enter a URL: [?](#)

Submit **Add Row**

Figure 13: Experimentálne dátá.

http://127.0.0.1:8000/en-us/data/predict/

MLTOX Language (en-us) Log in Sign up

[Home](#) > [Predict](#) > Results

Prediction results

1. SMILES: NC(Cc1cnc[nH]1)C(=O)O

Molecule model:

Descriptors:

Cycles: 1.0
Atom valence: 46.0
Positive sum: 1.6423568920798197
Hydrogen count: 9.0
CSP3: 0.3333333333333333

Result:

NPT
91.30%
Confidence

Figure 14: Návrhy užívateľa.

Submitted data

Molecule model:	Descriptors:	Submitted data:
<chem>OCC</chem>	Cycles: 0.0 Atom valence: 16.0 Positive sum: 0.43850220093603226 Hydrogen count: 6.0 CSP3: 1.0 Aromatic heavy atoms: 0.0 Hydrogen donors: 1.0 Hydrogen acceptors: 1.0	Phototoxicity: <input type="text" value="Non-phototoxic"/> Test used: <input type="text" value="3T3"/> Email: <input type="text" value="user@user.cz"/> URL: <input type="text"/>
		<input type="button" value="Validate"/> <input type="button" value="Delete"/>

2.SMILES: CCO

Molecule model:	Descriptors:	Submitted data:
<chem>OCC</chem>		

Figure 15: Niekolko vyhradených účtou potvrdzuje alebo maže návrhy.

4.3.3 Testování webové aplikace

Funkčnosť aplikacie jsme si ověřili díky sérii testů. Ty se soustřeďují především na dvě stěžejní části naší aplikace a to predikci fototoxicity a vytváření nového modelu.

Díky testům se zaručilo, že nepošleme do produkce kód, který by se později ukázal jako naprostě nefunkční. Krom lokálního testování je automaticky otestován každý požadavek na změnu v hlavní vývojové větvi. Pokud projde, tak se změnou hlavní větve se automaticky aktualizuje i produkční kód. Pokud naopak neprojde, produkční kód zůstává neměnný.

5 Používateľská príručka

Cieľom tejto príručky je jednoznačne vysvetliť postup používania výsledného produktu. Vzhľadom na to, že ide o webovú aplikáciu nasadenú na vzdialenom serveri, používateľ nemusí inštalovať nič dodatočné. Stačí tak mať nainštalovaný internetový prehliadač a ísť na webovú stránku <https://mltox.online>/

Zároveň treba dodať, že hoci predikovať fototoxicitu látky môže akýkoľvek neprihlásený používateľ (teda aj laik do chémie), rozšírená funkcia lita aplikácie (napr. odovzdávanie nových dát) je dostupná až po registrovaní a prihlásení sa do svojho účtu.

5.1 Domovská obrazovka

Úvodná obrazovka aplikácie, ktorá je prístupná všetkým používateľom. Obrazovka je minimalistická a obsahuje hlavne tlačidlo pre predikovanie a zopár informácií k projektu.

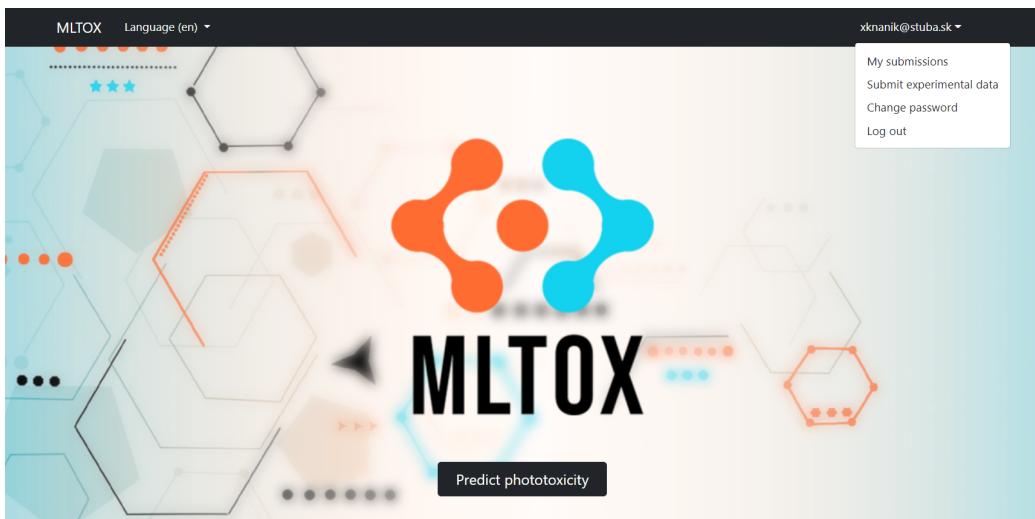


Figure 16: Domovská obrazovka s možnosťami zaregistrovaného používateľa.

Horná navigačná lišta je dostupná na každej obrazovke aplikácie a obsahuje možnosť zmeniť si jazyk a prihlásiť/registrovať sa. Prihlásený používateľ

má na výber ďalšie možnosti - môže si pozrieť zoznam svojich odovzdaných dát, môže nové dáta odovzdať, prípadne si zmeniť heslo a odhlásiť sa.

5.2 Registrácia/prihlásenie

Registrácia používateľa pozostáva z troch jednoduchých krokov: používateľ zadá svoju emailovú adresu, heslo a heslo znova.

Registrovať sa môžu len používatelia s emailovou adresou s autorizovanou doménou. Medzi autorizované domény patria viaceré univerzitné domény, pričom sa ich zoznam rozrastá. Ak Vaša doména nemá oprávnenie na vytvorenie účtu, odporúčame kontaktovať podporu.

The screenshot shows the 'Sign up' form on the MLTOX website. At the top, there is a dark header bar with the text 'MLTOX Language (en) ▾' on the left and 'Log in Sign up' on the right. Below the header is a light-colored sign-up form. The form has a title 'Sign up' at the top. A red error message box contains the text: 'Email domain is not authorized for sign-up! To authorize this email domain send an email to tim20_tp2021@gmail.com.' Below the message box are three input fields: 'Email' containing 'tester@gmail.com', 'Password', and 'Password repeat'. At the bottom of the form is a yellow 'Sign up' button. Below the button, a small note says 'Already have an account? [Log in](#)'.

Figure 17: Registrácia je povolená len pre určité domény.

Registrovaní používatelia sa môžu emailom a heslom prihlásiť. Ak ste heslo zabudli, môžete si ho kliknutím na odkaz s pomocou svojho emailu zmeniť.

5.3 Predikcia

Predikcia fototoxicity látok je dostupná pre všetkých používateľov, teda aj tých bez vytvoreného účtu. Používateľ môže predikovať viaceré molekuly

naraz. Pre predikovanie fototoxicity je potrebný len SMILES reťazec danej molekuly.

SMILES (akronym pre “Simplified molecular-input line-entry system”) je jednoduchý formát umožňujúci zápis chemickej štruktúry pomocou ASCII znakov. SMILES reťazce molekúl sú ľahko dohľadateľné na rôznych voľnodedostupných internetových chemických databázach.

Výsledky predikcie zahŕňajú výsledok predikcie a spoľahlivosť predikcie ako aj významné molekulové deskriptory a grafickú reprezentáciu molekuly. Ak je používateľ registrovaný, môže výsledok predikcie opraviť (ak disponuje dôkazom o (ne)phototoxicite).

The screenshot shows a web interface for predicting phototoxicity. At the top, there are navigation links: 'MLTOX' and 'Language (en-us)'. On the right, there is an email address: 'xmuskova@stuba.sk'. Below the header, the URL path is shown: 'Home > Predict > Results'. The main section is titled 'Prediction results' and contains the following information:

1. SMILES: CCO

Molecule model:	Descriptors:	Result:
<chem>CCO</chem>	Cycles: 0.0 Atom valence: 16.0 Positive sum: 0.43850220093603226 Hydrogen count: 6.0 CSP3: 1.0 Aromatic heavy atoms: 0.0 Hydrogen donors: 1.0 Hydrogen acceptors: 1.0	Test used: 3T3 Correct result: NPT Enter a URL: <input type="text" value="https://example.com"/> <input type="button" value="Submit"/> <input type="button" value="Cancel"/>

At the bottom of the main content area is a 'Back to predicting' button.

Figure 18: Používateľ vie zadať typ testu, výsledok a pridať experimentálne dáta na opravu.

5.4 Odovzdávanie nových dát

Prihlásený používateľ má možnosť pridávať nové údaje do systému, čím prispeje k lepšej úspešnosti modelu. Pri pridávaní dát používateľ zadá SMILES reťazec molekuly, typ vykonaného testu na phototoxicitu, výsledok testu a ak je možné, tak aj URL odkaz na informácie/zdroj, ktorý by výsledok testu podporil.

Možnosť zadať údaje do systému možno nájsť po prihlásení sa, v hornej lište, kliknutím na možnosť "Submit experimental data".

The screenshot shows a web-based application interface for submitting experimental data. At the top, there's a header bar with 'MLTOX' and 'Language (en-us)'. On the right side of the header, it says 'xmuskova@stuba.sk'. Below the header, the URL 'Home > Submit data' is visible. The main area is titled 'Submit experimental data'. It contains two rows for inputting data. Each row has three fields: 'SMILES' (containing 'CCO' and 'CO' respectively), 'Test Type' (dropdown menus set to '3T3' and '3D'), and 'Result' (dropdown menus set to 'NPT'). To the right of each row is a small red 'X' icon. Below these rows is a section labeled 'Enter a URL:' with a help icon (a question mark inside a circle). A text input field contains the URL 'https://example.com'. At the bottom of the form are two buttons: 'Submit' and 'Add Row'.

Figure 19: Odovzdávanie nových dát.

Údaje, ktoré sú zadávané do systému, musia byť najskôr overené autorizovanou osobou. Používateľ si môže pozrieť stav dát, ktoré odovzdal do systému. Každý príspevok má zobrazený status schválenia autorizovanou osobou: schválené, čakajúce na validáciu alebo zamietnuté.

Dáta, ktoré používateľ odovzdal, je možné pozrieť po prihlásení sa, v hornej lište, vybraním možnosti "My submissions".

5.5 Validovanie dát

Ofovzdané dáta je potrebné overiť. Pre aplikáciu MLTOX existuje možnosť pre vybraných používateľov stať sa autorizovanou osobou, ktorá môže schvalovať/zamietnať návrhy na nové dát.

Na stránke pre validáciu dát môže autorizovaná osoba vidieť návrhy na pridanie dát do systému. Každý návrh obsahuje údaje o fototoxicite molekuly spolu s vykonaným testom. Pre každý návrh sa môže overovateľ rozhodnúť, či návrh schváli alebo zamietne.

The screenshot shows the 'Submitted data' section of the MLTOX web application. At the top, there are dropdown filters for 'Phototoxicity' (set to 'Any') and 'Test used' (set to 'Any'), with a 'Search' button. Below this, the first entry is shown:

1.SMILES: CCO

Molecule model:	Descriptors:	Submitted data:
	Cycles: 0.0 Atom valence: 16.0 Positive sum: 0.43850220093603226 Hydrogen count: 6.0 CSP3: 1.0 Aromatic heavy atoms: 0.0 Hydrogen donors: 1.0 Hydrogen acceptors: 1.0	Phototoxicity: Non-phototoxic Test used: 3T3 Email: user@user.cz URL: Validate Delete

The second entry is partially visible:

2.SMILES: CCO

Molecule model:	Descriptors:	Submitted data:

Figure 20: Niekoľko vyhradených účtou potvrdzuje alebo maže návrhy.

V prípade, ak by ste sa chceli stať autorizovaným používateľom, kontaktuje prosím podporu.

PART II:

RIADENIE PROJEKTU

1 Úvod

V tejto časti si bližšie priblížime informácie o riadení projektu. Pozrieme sa na to, akým spôsobom sme organizovali riadenie úloh, aké techniky a prostriedky sme použili na riadenie a realizáciu projektu.

V tejto časti dokumentácie sa nachádzajú informácie o rolách členov tímu, aké manažérské pozície členovia tímu zastávali a aký je ich podiel práce na vytvorenom diele, dokumentácii a metodikách. Následne sa opisujú aplikácie jednotlivých manažmentov v projekte, ktoré sa odkazujú na príslušné metodiky. Súčasťou riadenia projektu je aj globálna retrospektíva a sumarizácia šprintov, v ktorých sa uvádza pokrok tímu na projekte.

2 Role členov tímu a podiel práce

2.1 Role členov tímu

Meno a Priezvisko	Rola
Matej Halinkovič	Architect & ML Engineer
Jakub Knánik	Web & QA Engineer
Jakub Maruniak	Data Scientist & ML Engineer
Kateřina Mušková	Data Scientist
Patrik Pavlačka	Web & ML Engineer
Tibor Sloboda	Domain Knowledge Expert

- **Matej Halinkovič**

Návrh ML modelov, návrh ML experimentov, návrh architektúry Data pipe, programovanie crawlerov chemických databáz, data science a návrh hlbokých neurónových sietí, Scrum Master, Dokument Master

- **Jakub Knánik**

Manažment kvality. Code review modulu Data pipe. Implementácia častí Django backendu a frontendu webapp, lokalizácia, implementácia testov, database engineering

- **Jakub Maruniak**

Rešerš dostupných zdrojov, analýza vhodnosti a extrakcia chemických látok z vedeckých publikácií

Počas letného semestra prerušil štúdium

- **Kateřina Mušková**

Analýza a čistenie dát (EDA), časť Cleaner, implementácia MLOps architektúry, database engineering, implementácia integračných a unit testov, UX testing, knowledge-base management

- **Patrik Pavlačka**

Vytvorenie a nasadenie webovej stránky tímu a jej pravidelné aktualizovanie, implementácia a redesign webového rozhrania, implementácia častí Django backendu, database engineering

- **Tibor Sloboda**

Analýza domény, výber a implementácia výpočtu chemických deskriptorov, data engineering, management verziovania, návrh architektúry MLOps; správa, návrh a implementácia CICD pipeline (deployment)

2.2 Podiel práce

2.2.1 Zimný semester

Šprint 1

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	17%
Jakub Knánik	20%
Jakub Maruniak	16%
Katerína Mušková	11%
Patrik Pavlačka	23%
Tibor Sloboda	13%

Šprint 2

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	17%
Jakub Knánik	19%
Jakub Maruniak	17%
Katerína Mušková	17%
Patrik Pavlačka	14%
Tibor Sloboda	16%

Šprint 3

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	11%
Jakub Knánik	17%
Jakub Maruniak	18%
Katerína Mušková	12%
Patrik Pavlačka	19%
Tibor Sloboda	23%

Šprint 4

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	22%
Jakub Knánik	14%
Jakub Maruniak	15%
Katerína Mušková	21%
Patrik Pavlačka	12%
Tibor Sloboda	16%

Šprint 5

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	16%
Jakub Knánik	13%
Jakub Maruniak	16%
Katerína Mušková	21%
Patrik Pavlačka	20%
Tibor Sloboda	14%

2.2.2 Letný semester

Šprint 6

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	20%
Jakub Knánik	20%
Jakub Maruniak	0%
Katerína Mušková	20%
Patrik Pavlačka	20%
Tibor Sloboda	20%

Šprint 7

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	18%
Jakub Knánik	21%
Jakub Maruniak	0%
Katerína Mušková	22%
Patrik Pavlačka	17%
Tibor Sloboda	22%

Šprint 8

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	21%
Jakub Knánik	19%
Jakub Maruniak	0%
Katerína Mušková	21%
Patrik Pavlačka	23%
Tibor Sloboda	16%

Šprint 9

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	19%
Jakub Knánik	22%
Jakub Maruniak	0%
Katerína Mušková	17%
Patrik Pavlačka	19%
Tibor Sloboda	23%

Šprint 10

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	22%
Jakub Knánik	18%
Jakub Maruniak	0%
Katerína Mušková	20%
Patrik Pavlačka	19%
Tibor Sloboda	21%

Šprint 11

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	20%
Jakub Knánik	20%
Jakub Maruniak	0%
Katerína Mušková	20%
Patrik Pavlačka	20%
Tibor Sloboda	20%

Šprint 12

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	19%
Jakub Knánik	20%
Jakub Maruniak	0%
Katerína Mušková	21%
Patrik Pavlačka	22%
Tibor Sloboda	18%

Šprint 13

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	21%
Jakub Knánik	17%
Jakub Maruniak	0%
Katerína Mušková	18%
Patrik Pavlačka	21%
Tibor Sloboda	23%

Šprint 14

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	22%
Jakub Knánik	22%
Jakub Maruniak	0%
Katerína Mušková	21%
Patrik Pavlačka	16%
Tibor Sloboda	19%

Šprint 15

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	18%
Jakub Knánik	22%
Jakub Maruniak	0%
Katerína Mušková	20%
Patrik Pavlačka	21%
Tibor Sloboda	19%

Šprint 16

Meno a Priezvisko	Percentuálny podiel
Matej Halinkovič	20%
Jakub Knánik	19%
Jakub Maruniak	0%
Katerína Mušková	20%
Patrik Pavlačka	22%
Tibor Sloboda	19%

Práca na milníku

Autor	Časti milníka
Matej Halinkovič	Celkový pohľad na systém, Moduly systému, Role členov tímu a podiel práce, Manažment a metodológia správy úloh, Manažment a metodológia dátovej vedy
Jakub Knánik	Úvod - Inžinerske dielo, Riadenie projektu Globálne ciele projektu, Manažment a metodológia kvality, Manažment a metodológia verziovania projektu, Prihláška na TP Cup, Používateľská príručka
Kateřina Mušková	Globálna retrospektíva, Sumarizácia šprintov, Manažment a Metodológia komunikácie, Manažment a Metodológia ukladania informácií
Tibor Sloboda	Moduly systému, Používané chemické deskriptory
Patrik Pavlačka	Webová aplikácia

3 Aplikácie manažmentov

3.1 Manažment kvality

Súčasťou projektu je aj zabezpečovanie kvality kódu, a to prostredníctvom dodržiavania zaužívaných štandardov a nami ustanovených pravidiel. Vzhľadom na povahu projektu a možnosti jeho ďalšieho rozširovania si myslíme, že štandardy kvality prispejú k lepšej a rýchlejšej pochopiteľnosti kódu. Taktisto udržiavaním kvality cielime na modulárnosť, ktorá pomôže k dosiahnutiu nižšej komplexnosti kódu a lepšej schopnosti rozširovania funkcia-

systému.

Na overenie dodržanej kvality slúži code review, počas ktorého sa posudzuje kvalita kódu z hľadiska chybovosti a dodržania štandardov. Konkrétnie informácie o písaní kódu podľa priatých pravidiel možno nájsť v prílohe A Metodika zabezpečovania kvality kódu.

3.2 Manažment testovania

Testovanie softvéru je jednou z nevyhnutných zložiek zabezpečovania kvality softvéru, a preto určité zaužívané spôsoby testovania aplikujeme aj pre tento projekt. Cieľom testov je nielen odhalovanie už prítomných defektov, ale aj zabezpečovanie správnosti funkcia-

lít. Keď sa kód určitej funkcie pokrytej testom zmení, test by mal byť schopný odhaliť, či zmena kódu nespôsobila chybu vo funkčnosti daného kódu.

Je vysoko odporúčané testovať inkrementálne, keďže vieme problémy zachytiť postupne a v skorších fázach vývoja. Pravidlá a konkrétnosti implementovania testovania v projekte možno nájsť v prílohe B.

3.3 Manažment verziovania projektu

Stav jednotlivých častí projektu môže byť rôznorodý, a preto je vhodné používať systém verziovania projektu, ktorý nám umožní zachovávať si rôzne verzie projektu. Takto budeme môcť pracovať paralelne na viacerých funkcia-

lítach a podsystémoch, čím zvýšime našu efektivitu.

Pre potreby verziovania používame distribuovaný systém riadenia revízií Git, ktorý je napojený na online nástroj pre verziovanie kódu Github. Konkrétnie informácie o štruktúre projektu a jeho verziovaní možno nájsť v prílohe C Metodika verziovania kódu.

3.4 Manažment správy úloh

Pre vytváranie, sledovanie a rozdelenie úloh medzi členov tímu využívame nástroj Jira. Má rozsiahlu funkciu, ktorá nám zjednoduší udržiavanie prehľadu o tom, ktorý člen tímu pracuje na čom, sledovač ako napredujeme v jednotlivých častiach projektu. Presná metodika našej práce s týmto nástrojom je popísaná v prílohe E.

3.5 Manažment dátovej vedy

Náš projekt sa z veľkej časti zamierava na dátovú vedu, pre ktorú klasický Agile prístup nie je ideálny. Z tohto dôvodu sme naše postupy rozšírili o metodológiu CRISP-DM, ktorá je bližšie popísaná v prílohe F.

3.6 Manažment komunikácie

Keďže zadaný projekt nie je individuálny, ale skupinový, je nutné sa dohodnúť na spôsobe komunikácie. Bolo nutné zvoliť príslušné kanály, a spôsob akým budú použité. Keďže sa v novembri zhoršila pandemická situácia, všetka komunikácia sa presunula do online prostredia, čím manažment komunikácie nabral na dôležitosť. Podrobnejšie informácie o zvolenej metodológii možno nájsť v prílohe H

4 Globálna retrospektíva

4.1 Zimný semester

Na začiatku cesty nášho projektu sa bolo treba najskôr zorientovať v téme fototoxicity. Okolo chémie sa točila diskusia nielen v tíme, ale aj našich stretnutiach s vedúcou tímu. Po aspoň čiastočnom zorientovaní v odbore sme začali vytvárať prvé modely, ktoré sme menili v nadväznosti na ďalšie zistenia a nové dáta.

Kedže by aplikácia mala byť pohodlne použiteľná užívateľom, je súčasťou nášho projektu aj vybudovanie webovej aplikácie, ktorou sme sa začali zoberať v treťom špriente. Zatiaľ sme sa sústredili predovšetkým na návrh kvalitnej architektúry s možnosťou autentifikácie. Podrobnejší prehľad činností je potom spísaný v nasledujúcej kapitole 5.

4.2 Letný semester

Počas letného semestra sme sa sústredili najmä na uvedenie nášho produktu do produkcie. Podarilo sa nám vytvoriť web aplikáciu, ktorá je (aspoň aktuálne) zadarmo dostupná na deskriptívnej, ľahko zapamäteľnej doméne mltox.online. Produkt, ktorý sme vytvorili je ľahko použiteľný a splňa požiadavky stanovené našou cielovou skupinou používateľov - toxikológmi.

Venovali sme sa ďalej aj experimentovaniu s cielom vytvoriť čo najúspešnejší model pre predikciu fototoxicity. V tomto smere sme boli tiež úspešní a podarilo sa nám vytvoriť model, ktorý porazil súčasný stav poznania a je najlepší na trhu. Na základe týchto výsledkov sme sa rozhodli zúčastniť niekoľkých konferencí a naše výsledky publikovať aj ako vedecký článok.

5 Sumarizácie šprintov

5.1 Zimný semester

Šprint č. 1

Cieľom nášho prvého šprintu bolo v prvom rade oboznámiť sa s problematikou fototoxicity, aby sme dobre chápali dátam s ktorými budeme pracovať, t.j. formy zápisov a reprezentácie molekúl pomocou rôznych deskriptorov. Preštudovali sme poskytnutú literatúru, a tam kde to bolo možné extrahovali dostupné chemické látky, relevantné k nášmu problému. Z nich sme pomocou webových databáz chemických látok získali deskriptory, nad ktorými sme vykonávali prieskumnú analýzu. Samozrejme, okrem práce s dátami sme spojazdnili webovú stránku s prezentáciou nášho projektu, nasetupovali GitHub, JIRA a Confluence projekty.

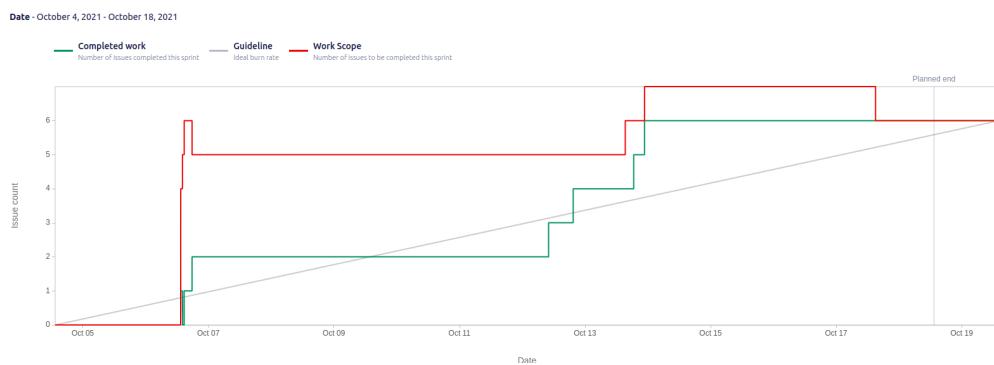


Figure 21: BurnUp graf šprintu č.1

Čo sme spravili dobre?

- celkovo rýchla odozva nášho tímu a nastavenie organizačných produktov včas nám pomohlo udržiavať si rozhľad o tom, na čom je momentálne dôležité pracovať
- pripravili sme rozličné komunikačné nástroje, ktoré značne zjednodušujú spoluprácu

Čo nám robilo problémy?

- charakter projektu a nami definované počiatočné úlohy spôsobili, že identifikovať a následne priradiť jednotlivé role členom tímu, bolo náročnejšie ako sme predpokladali
- bez priradených rolí nastávali problémy s prekrývaním činností
- počiatočné uchopenie problematiky, nakoľko sa jedná o rozsiahlu tému, o ktorej sme mali doposiaľ len okrajové vedomosti
- pre danú problematiku je náročné získať dostatočné množstvo dát

Návrhy na zlepšenie

- pri plánovaní taskov je potrebné na začiatku zohľadniť závislosti voči iným, aby sme naplno využili potenciál a neboli počas šprintu blokovaní
- pre udržanie prehľadnosti by bolo možno dobré k taskom v JIRE počas ich riešenia a po dokončení pridávať stručné komentáre, prípadne súbory alebo linky na confluence. Všetci členovia tímu tak budú lepšie informovaní o stave jednotlivých úloch a ich výsledkov
- streamlinovanie komunikácie a činností na jednotlivých stretnutiach pre lepšie zúžitkovanie relatívne limitovaného času

Šprint č. 2

V rámci tohto šprintu bola zavedená implementácia rozvinutých modelov pre predikciu a vyriešenie problému získavania nových dát. Podarilo sa nám implementovať nové systémy, ktoré nám do budúcnosti ušetria viac času. Pokročili sme vo všetkých smeroch, či už v rámci riadenia projektu, ale aj samotného vývoja - data engineering a modelling. Začali sme tiež plánovať štruktúru finálnej webovej aplikácie. Zároveň sme narazili na niekoľko problémov, primárne stále zápasíme s dátovou doménou, či už s množstvom dát, prístupu k nim, alebo vhodnosťou rôznych deskriptorov.

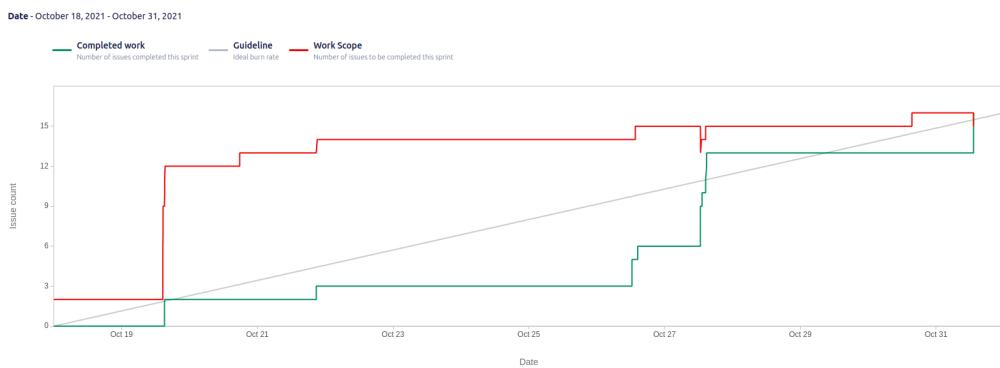


Figure 22: BurnUp graf šprintu č.2

Čo sme spravili dobre?

- Implementovali sme nové postupy pre prácu s Jira, ktoré nám umožňujú lepšie spolupracovať na úlohách a udržiavať tak lepší prehľad o tom čo robia jednotliví členovia tímu
- Navrhli sme pipeline pre náš proces spracovania dát, ktorá nám umožňuje automatizovať väčšinu práce pre získanie nových informácií a teda nám v budúcnosti ušetrí veľa času
- Podarilo sa nám nájsť spôsob ako vypočítať nové relevantné deskriptory, ktoré neboli dostupné v databázach, s ktorými pracujeme
- Prihlásili sme sa do TP Cupu.

Čo nám robilo problémy?

- Stále sa nám nepodarilo vyriešiť nedostatok dát, najmä pre nefototoxicke látky
vypĺýva z toho problematické vyhodnotenie úspešnosti modelu. Náš prístup je úspešný na našom datasete, ale nevieme spoľahlivo predpovedať jeho úspešnosť v reálnom svete
- Zistili sme, že žiadny z členov tímu nemá zatiaľ skúsnosti s deploymentom modelov pre web aplikácie.
- Crawl dát z verejných stránok spôsobil, že sme dostali ban na zopár IP adres pre isté webstránky

Návrhy na zlepšenie

- Zistiť či sa dajú vypočítať všetky potrebné deskriptory zo SMILES kódu, ak áno, tak implementovať systém, ktorý to dokáže zvládať v reálnom čase
znížime tak závislosti na rôznych verejných databázach, ktoré by bolo problematické používať hlavne vo finálnom produkte
- Lepšie porozumieť chemickému pozadiu a princípom, ktoré spôsobujú fototoxicitu, aby sme mohli vytvoriť robustnejší a spoľahlivejší model
- Vyriešiť problém s nedostatkom dát
- Doučiť sa prístupy v web deploymentu modelov a začať pracovať na aplikačnej stránke nášho projektu

Šprint č. 3

Hlavným cieľom tohto šprintu bolo vytvoriť prototyp webovej aplikácie pre ML Model Deployment a extrahovať ďalšie dátu z článkov a databáz. Začali sme pracovať aj na dokumente Míľnik 1 (prvé 3 šprinty) a návrhu architektúry webovej aplikácie. Počas tohto šprintu sme mali aj hostovanú prednášku s Dr. Helenou Kandárovou. Dozvedeli sme sa nové informácie o fototoxicite, aké sú charakteristické vlastnosti fototoxických látok, kedy je potrebne testovať fototoxicitu látky, aké testy sa v súčasnosti robia na fototoxicitu a zvalidovali sme nami vybrané deskriptory.

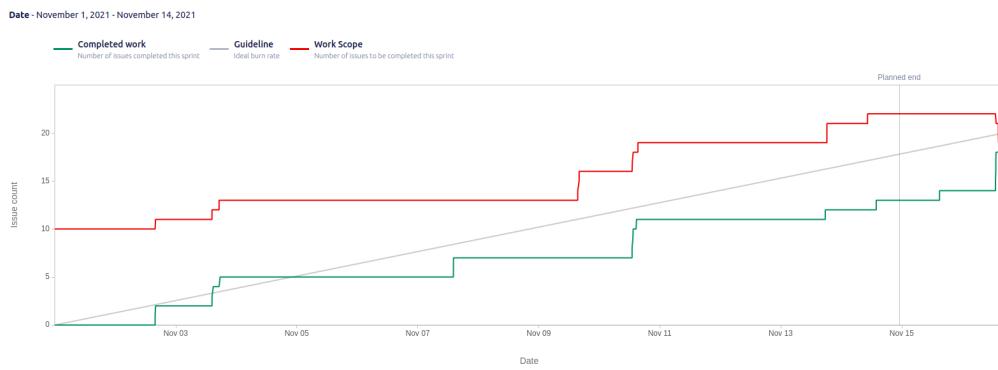


Figure 23: BurnUp graf šprintu č.3

Čo sme spravili dobre?

- Podarilo sa nám získať viac dát o fototoxických a nefototoxických látkach z článkov a databáz
- Implementovali sme prototyp webovej aplikácie s využitím webového frameworku Flask
- Dokončili sme našu data pipelinu
- Získali sme nové deskriptory a pretrenovali sme náš model na nich
- Prehĺbili sme si naše znalosti z oblasti phototoxicity

Čo nám robilo problémy?

- Stále máme pomerne malé množstvo dát
- S niektorými látkami z japonskej databázy bol problem identifikovať, či sú skutočne nefototoxicke na základe ich absorpcie svetla, bude potrebné overiť tieto látky, či sú skutočne nefototoxicke

Návrhy na zlepšenie

- Získať ďalšie dáta o fotoxických a nefototoxicých látkach
- Overiť dáta z japonskej databázy a zistiť, ktoré látky sú fototoxicke/nefototoxicke
- Dopracovať návrh architektúry webovej aplikácie a jej komunikácie s ML modelom

Šprint č. 4

Prioritou tohto šprintu bolo spísanie prvého miľníku, čo sa odráža aj v úlohách z tohto šprintu. Okrem písania samotného dokumentu však z tvorby miľníku vyplynuli aj úlohy na dizajn architektúry pre našu budúcu webovú aplikáciu. Identifikovali sme potrebné časti ako ML pipe pre aplikáciu, databáza predikcií a rozhranie pre overovanie identity používateľa. Architektúru daných častí sme následne navrhli príslušnými diagramami. Tiež sme ďalej pokračovali v zbieraní ďalších údajov o fotoxicite látok.

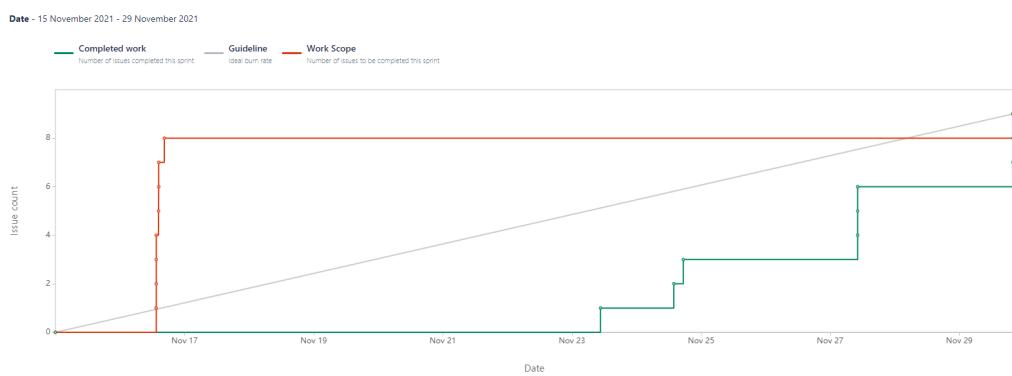


Figure 24: BurnUp graf šprintu č.4

Čo sme spravili dobre?

- Podarilo sa nám získať viac dát o fotoických a nefototoxických látkach z článkov a databáz
- Vytvorili sme (snáď) kvalitný dokument pre prvý miľník, ktorý nám pomôže aj v neskorších fázach projektu
- Prediskutovali a navrhli sme robustnú architektúru pre web aplikáciu, ktorá by mala zahŕňať všetky požiadavky a zjednodušiť tak jej budúcu implementáciu

Čo nám robilo problémy?

- Stále máme pomerne malé množstvo dát
- Tvorba dokumentov je zdĺhavá a zvykne nám zabrať viac času ako očakávame
- Dostávame sa do fázy, kde začíname vyčerpávať možnosti ako pokročiť

Návrhy na zlepšenie

- Získať ďalšie dáta o fototoxickejch a nefototoxickejch látkach
- Dôkladne premyslieť postup v najbližších šprintoch, aby sme sa nezamotali vyvýjaním vecí, z ktorých sa stanú slepé uličky

Šprint č. 5

Riešili sme spôsob validácie používateľov, návrh backendu a používateľského rozhrania. Zároveň sme skúmali možnosti, ako prepojiť Django s JavaScriptom. V druhej časti šprintu sme navštívili chemický ústav na SAV. Pri exkurzii sme sa zoznámili s experimentami, pri ktorých sa zistuje fototoxicity a určuje absorpčné spektrum látok, ktoré s fototoxicitou súvisia.

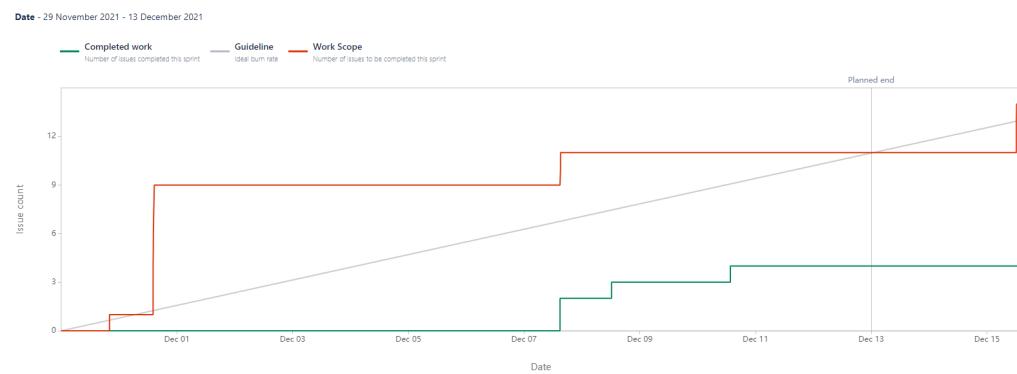


Figure 25: BurnUp graf šprintu č.5

Čo sme spravili dobre?

- Víme, jakým způsobem v praxi probíhají chemické experimenty, které jsou pro nás relevantní .
- Naučili jsme se pipetovat. Pokud se do budoucna ukáže IT jako neperspektivní oblast, budeme se moci zaměstnat jako pomocná síla na SAV.
- Rozchodili jsme první prototyp web aplikace.

Čo nám robilo problémy?

- Lehké problémy nám dělala komunikační technika.
- Správný způsob autorizace autentifikace a autorizace uživatele.

Návrhy na zlepšenie

- Vylepšit bezpečnost aplikace.

5.2 Letný semester

Na základe späťnej väzby zo stretnutia o RTP medzi semestrami sme upravili priebeh šprintov voči minulému semestru. Skrátili sme dobu trvania šprintov na jeden týždeň, čo nám umožnilo oveľa dynamickejšie reagovať na vývoj situácie, s čím sme mali v minulom semestri problém.

Tiež sme implementovali systém story pointov. Najčíslnejšia úloha v šprinte dostala vždy 1 SP, ostatné úlohy sme odhadovali ako násobok náročnosti danej úlohy. Pomohlo nám to udržať konšantnosť naprieč šprintami, nakoľko najčíslnejšie úlohy boli vždy písanie zápisov zo stretnutí a retrospektív predchádzajúceho šprintru.

Vizualizácie v grafoch stále netvoria ideálny pohľad na náš progres v tíme, nakoľko SP pre niektoré úlohy boli dohodnuté až po začatí šprinut a využívanie stavu VALIDATION často oneskoruje pridanie hotových SP do grafu.

Šprint č. 6

Úvodný šprint letného semestra sme zahájili analýzou súčasného stavu projektu. Identifikovali sme potrebu nasadenia prototypu stránky na VM, aby sme ho mohli prezentovať verejnosti a začali sme implementovať nový dizajn aplikácie.

Dr. Prnovej sa podarilo získať licenciu pre SW Alvadesc, ktorý nám umožní získať väčšie množstvo informácií o molekulách a má tak potenciál zlepšiť výkon nášho modelu.

Z organizačného hľadiska sa snažíme implementovať zmeny, o ktorých sme diskutovali pri RTP konzultácii. Začíname tento semester so snahou pracovať v jednotyždňových šprintoch namiesto dvoj týždňových pre lepší manažment externých faktorov. Zaviazali sme sa pre každý task zaviesť Definition of Done a implementovali sme systém Story pointov.

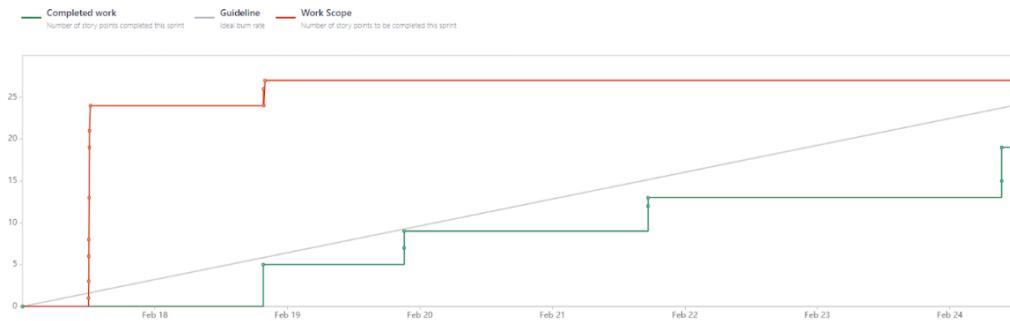


Figure 26: BurnUp graf šprintu č.6

Čo sme spravili dobre?

- Priestor pre zlepšenie predikčného modelu - implementácia novej dátovej sady, experimenty s NN.
- Začiatky implementácie redizajnu
- Organizačné zmeny riadenia, veríme že povedú k efektívnejšej spolupráci

Čo nám robilo problémy?

- Momentálnym hlavným bodom úrazu je neskúsenosť členov tímu s web developmentom a nasadzovaním stránok. Narazili sme na problémy s paralelným manažmentom tímovej stránky a web aplikácie, nepodarilo sa nám nasadiť Django server.

Návrhy na zlepšenie

- Zvyknúť si na novo implementované riadenie.
- Kontaktovať ľudí pre pomoc s nasadzovaním webu, aby členovia tímu nestrácali zbytočne veľa času.
- Doriešiť systém autentifikácie používateľov web aplikácie.

Šprint č. 7

Ústredným problémom druhého šprintu bolo nasadenie webovej aplikácie. Podarilo sa ju spustiť na rovnakej VM akú má aj stránka tímu, hoci do budúcnosti by bolo vhodné web aplikáciu osamostatniť. Riešili sme tiež autentifikáciu používateľov, kde sme sa rozhodli rozdeliť používateľov na dve skupiny: základných a manuálne schválených (ktorí majú väčšie právomoci), s čím súvisí aj vlastný SMTP server.

V priebehu šprintu prebehla diskusia o dôležitosti deskriptorov z Alvadesc SW, zároveň sme však boli dočasne zastavení podmienkami použitia tohto softvéru. V modelovaní sme sa posunuli k lepším výsledkom vďaka MLP a skúmali sme možnosť použiť NN ako alternatívu.

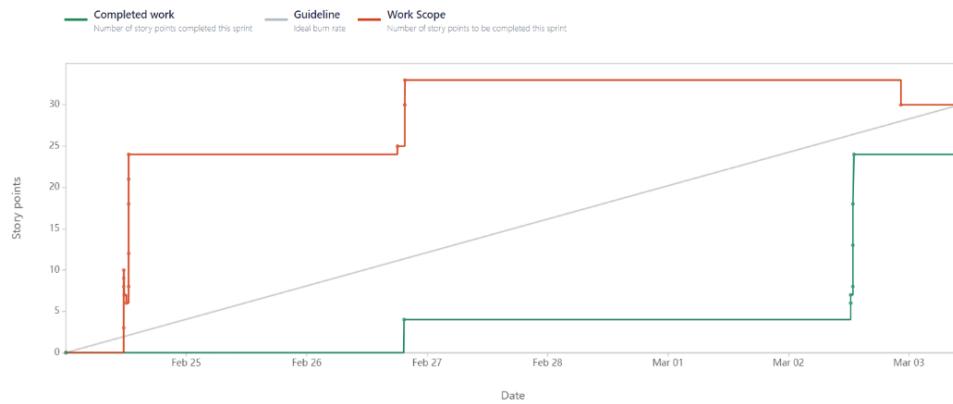


Figure 27: BurnUp graf šprintu č.7

Čo sme spravili dobre?

- Nasadenie webovej aplikácie.
- Preskúmali sme rôzne možnosti modelovania (MLP, NN).

Čo nám robilo problémy?

- Nedostatočná skúsenosť s web developmentom sa zdá aj naďalej byť najväčšou prekážkou.

Návrhy na zlepšenie

- Získanie novej domény pre web aplikáciu.
- Implementovanie CI/CD pre web aplikáciu.
- Oboznámiť sa s tvorbou účtov a autentifikáciou v Djangu.

Šprint č. 8

V rámci šprintu sa nám podarilo spraviť pokroky v oblasti používateľských účtov spravením funkcionality pre vytváranie používateľských účtov za pomocí Djanga. Našu aplikáciu bude voľne dostupná pre všetkých, avšak používatelia z overených vedeckých inštitúcií budú mať zvýšenú funkcia-

itu.
Pre náš existujúci dataset sme vygenerovali MOL súbory, aby mohli byť použité v SW Alvadesc, predtým ako vyriešime problémy s licencovaním.

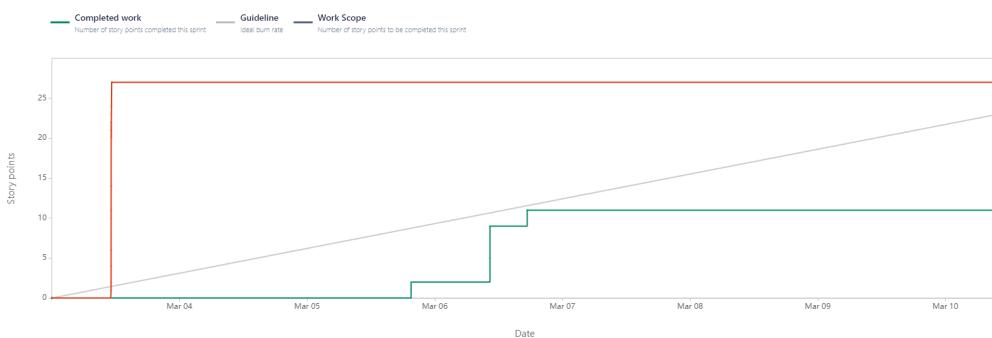


Figure 28: BurnUp graf šprintu č.8

Čo sme spravili dobre?

- Vytvorenie registračného systému
- Implementovanie systému akademických domien
- Získanie vlastnej domény pre aplikáciu - mltox.online

Čo nám robilo problémy?

- Nedoriešený stav Alvadesc licencie
- Potrebujeme vyriešiť systém pre pridávanie akademických domén, nakoľko ich manuálne pridávanie je najmä do budúcnosti nepraktické

Návrhy na zlepšenie

- Dokončiť CI/CD pre web aplikáciu.

Šprint č. 9

Hlavné zložky tohto sprintu tvorila automatizácia, používateľské účty a deskriptory z Alvadesc. Vytvorenie CI/CD pipeline bolo pre nás dôležité, vzhľadom na to, že už disponujeme nasadenou webovou aplikáciou na serveri a chceli sme, aby sa vytvorené a overené zmeny automaticky prejavili na serveri. Takisto práve zabezpečujeme testovanie funkcionality a automatické overenie pridaného kódu do repozitára testami nám povie, či nám zmena v kóde nezničila správnu funkcionality.

Okrem toho sme vytvorili možnosť registrácie/prihlásenia s podporou Django frameworku. Využitie softvéru tretej strany pre tieto účely nám výrazne pomohlo, vzhľadom na to, že správu a bezpečnosť účtov sme nemuseli implementovať sami.

Taktiež sme získali ďalšie informácie o molekulách z Alvadesc SW a skúšali sme modelovať aj s použitím deskriptorov z Alvadesc. Museli sme sa tu však vysporiadať s problémom s licenciou - licencia viazaná na HW.

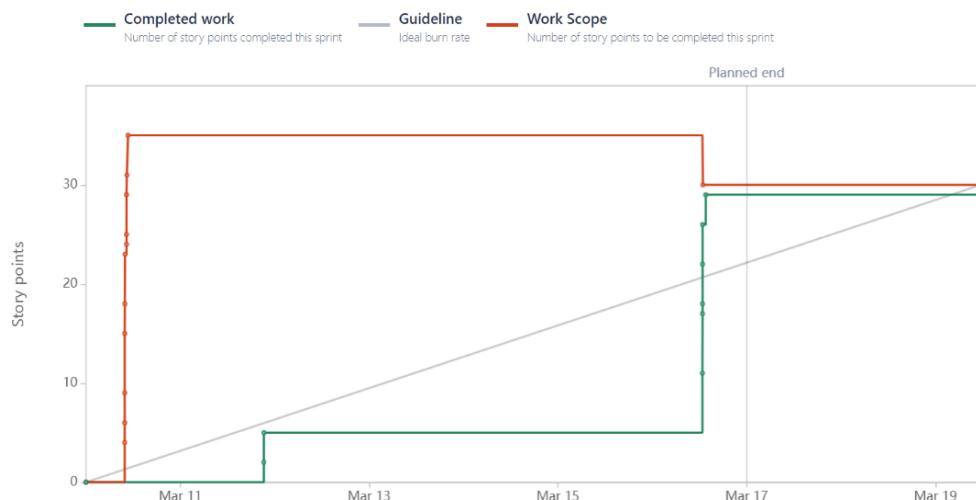


Figure 29: BurnUp graf šprintu č.9

Čo sme spravili dobre?

- CI/CD - po zmene v hlavnej vetve repozitára webovej appky sa automaticky spustia testy. Zároveň je vetva prepojená s automatickým

nasadením, a teda zmeny sa priamo prejavia na web appke na serveri.

- Využili sme možnosti Django frameworku, čím sme bez nutnosti vlastnej implementácie vedeli použiť viaceré funkcionality (napr. v oblasti správy účtov a security).

Čo nám robilo problémy?

- Problém s presúvaním licencie na Alvadesc medzi rôzny HW, je potrebné to odkomunikovať s Alvadesc. Počas šprintu sme to čiastočne vedeli obísť použitím HW na ktorom už licencia na Alvadesc bola.

Návrhy na zlepšenie

- Data augmentation pre vylepšenie modelovania.
- Do ďalších šprintov vzniká potreba začať tvoriť materiály pre TP Cup (IIT.SRC paper).

Šprint č. 10

Ústrednou témou tohto šprintu bolo napísanie článku pre IIT.SRC, vzhľadom na blížiaci sa marcový deadline. Väčšinu obsahu článku sa podarilo napísať už v tomto šprinte: v článku sa opísal problém fototoxicity látok, spôsoby testovania látok, dôležité chemické deskriptory, opísal sa proces modelovania ako aj pipeline pre machine learning.

Kľúčovým momentom v tomto šprinte bolo nájdenie článku so SOTA riešením predpovedania fototoxicity pomocou strojového učenia, v ktorom sme tiež našli ďalšie množstvo nových látok. Pomocou nášho modelu a s využitím RDKit deskriptorov dokázal náš model mať o 2% lepšiu senzitivitu a 5% lepšiu špecifickitu ako spomínané SOTA riešenie.

Okrem iných vecí sme tiež riešili jednotkové testy ako aj časť stránky, na ktorej môže oprávnený používateľ odsúhlasiť ešte neoverené odovzdané dáta o fototoxicite nových látok. Okrem toho sme sa venovali možnému spôsobu augmentácie dát.

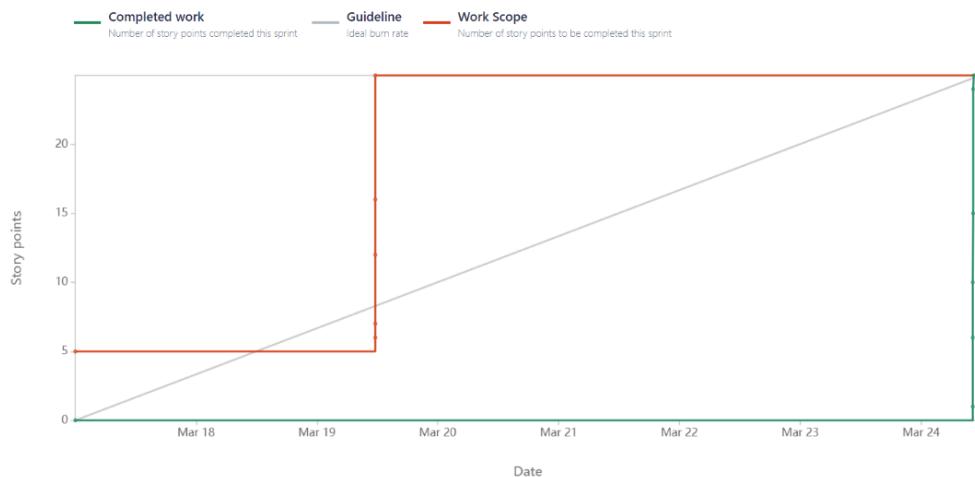


Figure 30: BurnUp graf šprintu č.10

Čo sme spravili dobre?

- Napísanie podstatnej časti článku pre konferenciu IIT.SRC.
- Nájdenie článku so SOTA riešením - lepšie výsledky nášho modelu.

Čo nám robilo problémy?

- Augmentácia dát - problém hľadania dvojice bodov s najkratšou vzdialenosťou.

Návrhy na zlepšenie

- Dopísať článok na IIT.SRC, pridať kapitolu k nasadeniu modelu.
- Zjednotiť jazyk na stránke (resp. spraviť lokalizáciu).

Šprint č. 11

V tomto šprinte bolo potrebné dokončiť článok na IIT.SRC. Do článku sme pridali ešte niektoré časti (ako napr. kapitola k nasadeniu modelu) a článok sme odovzdali.

V rámci webovej aplikácie sme implementovali preklad textov, ktorý umožňuje Django. Takýmto spôsobom vieme zabezpečiť vytvorenie preklad označených textov. Momentálne ponúkame anglickú a slovenskú lokalizáciu.

Súčasťou postupu na webovej aplikácii v tomto šprinte bolo tiež implementovanie backendu pre validáciu odovzdaných látok. Používateľmi odovzdané látky separátne uložené ako “suggestions” vie odobriť oprávnený používateľ (treba do ďalšieho šprintu implementovať účty s daným oprávnením - “chemist”).

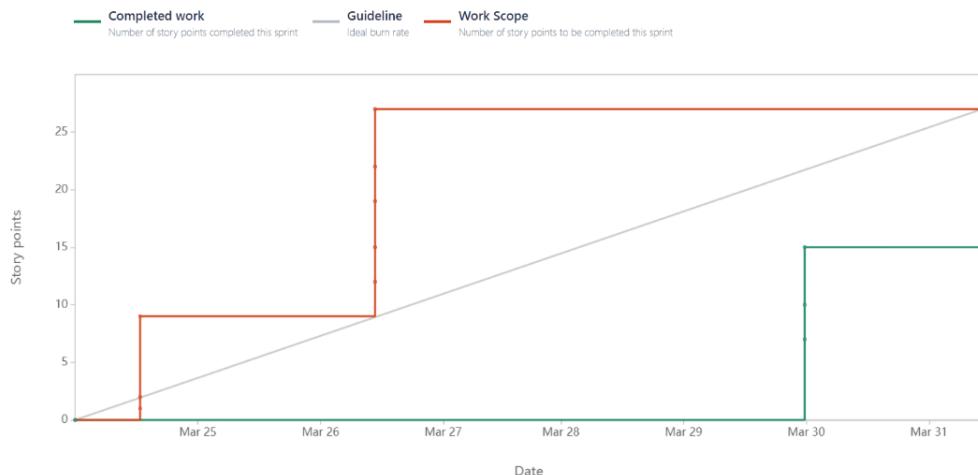


Figure 31: BurnUp graf šprintu č.11

Čo sme spravili dobre?

- Plán publikovať v žurnále Toxicology in Vitro.

Čo nám robilo problémy?

- Test suite je potrebné opraviť.

Návrhy na zlepšenie

- MLops pre automatické pretrénovanie modelu podľa nových dát.

- Zakomponovanie účtu “chemik” a následné zapojenie vedúcej tímu ako prispievateľa dát do web aplikácie.

Šprint č. 12

V tomto šprinte sme znova o niečo viac vylepšili webovú aplikáciu, tentoraz sme zabezpečili validáciu používateľského vstupu a implementovali sme účty s oprávnením na úrovni “chemist”, ktoré umožňuje používateľovi schvaľovať/zamietnuť návrhy pridania chemických látok do databázy.

V modelovaní sme implementovali modul na zrýchlenie procesu experimentovania s hľadaním najlepšieho modelu. Pokračovali sme v eksperimentoch a zároveň sme zväžili možnosť použitia grafových neurónových sietí, ktoré by mohli pre nás problém dať kvalitnejšie výsledky.



Figure 32: BurnUp graf šprintu č.12

Čo sme spravili dobre?

- Účty pre chemikov nám umožňujú zahrnúť vedúcu tímu do procesu testovania webapppky.
- Opravili sme chyby v testoch.

Čo nám robilo problémy?

- Hoci sme za predošlé šprinty písali zápisnice a retrospektívy, mierne sme sa pozabudli v ich updatovaní na stránke tímu, čo sme teraz napravili.

Návrhy na zlepšenie

- Začať sa v ďalšom šprinte venovať ďalším úlohám spojeným s TPCupom, teda video a poster na IIT.SRC.
- Preskúmať, implementovať a otestovať grafové neurónové siete.
- Preskúmať a vylepšiť dizajn/UX webappky.

Šprint č. 13

Blížiaci sa koniec semestra nás donútil začať pracovať na zvyšku výstupov pre TP cup. Prvým krokom bol návrh videa a zároveň sme začali s návrhom plagátu na IIT.SRC Začali sme tiež pracovať na zlepšovaní UX našej webovej aplikácie.

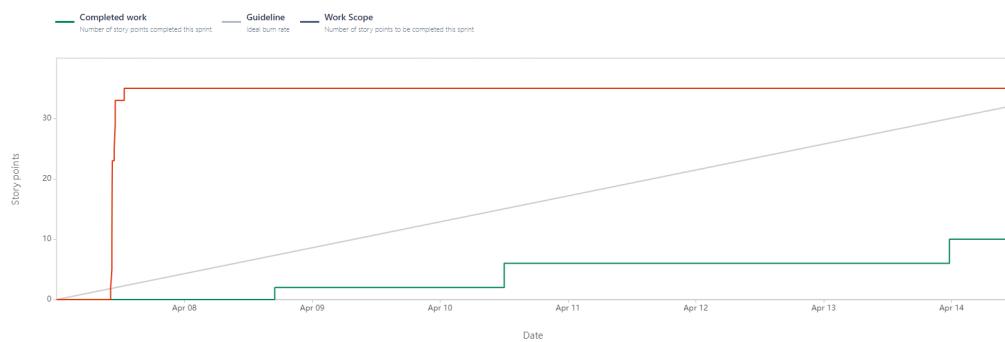


Figure 33: BurnUp graf šprintu č.13

Čo sme spravili dobre?

- Dizajn super videa pre TP Cup

Návrhy na zlepšenie

- Zistili sme viaceré nedostatky s úrovňou UX našej web aplikácie
- Potrebujeme dedikovať viac času pre GNN experimenty

Šprint č. 14

Úspešně jsme dokončili poster a většinu příprav na ITSRC. Připravujeme se na další konference. MLOps je taktéž v plném provozu.

Podařilo sa nám úspešne rozbehnuť prvé experimenty s grafovými neurónkami, a dosiahnuť ešte lepšie výsledky než s RandomForestom. Treba ďalej experimentovať a vylepšovať model, má potenciál.

Stránka dostala ďalšie malé vylepšenia, a fixli sme pári bugov a spadnutú predikciu. Malo by to už byť OK.

Video sme spravili, a na postery sa pracuje. Treba popohnať ten poster a blogpost, ale zatiaľ máme kvalitné materiály, len to treba ďalej rozvíjať.

MLOps je na dobrej ceste a už sa implementovalo automatické pretrénovanie, ešte treba pridať testy a doladiť pári detailov, najmä ohľadom manipulácie dát.



Figure 34: BurnUp graf šprintu č.14

Čo sme spravili dobre?

- Pokročili jsme s vývojem modelů (GNN, XGBoost)
- Opět jsme vylepšili uživatelskou přívětivost stránky, UX
- Dokončili jsme ITSRC poster

- Mimo technických hard skills rozvíjíme i naše literární cítění a na meetingu jsme recitovali Mor ho! Od Sama Chalupky.

Čo nám robilo problémy?

- Narůstající práce přímo související s vývojem - okolo článků, posterů, přihlášek, ...

Návrhy na zlepšenie

- Rozložit psaní na více lidí do budoucna.

Šprint č. 15

Tento šprint sa vyznačuje svojou rôznorodosťou. Pokračovali sme vo vylepšovaní produktu vo všetkých smeroch, pracovali sme na výstupoch pre TP Cup (video, blogpost), riešili sme účasť na konferenciach ESTIV a EuroQSAR a zlepšovali sme UX nášho produktu.

Napredujú aj experimenty s grafovými neurónovými sieťami, avšak pre nedostatok času na výskum budú finalizované až po skončení semestra.

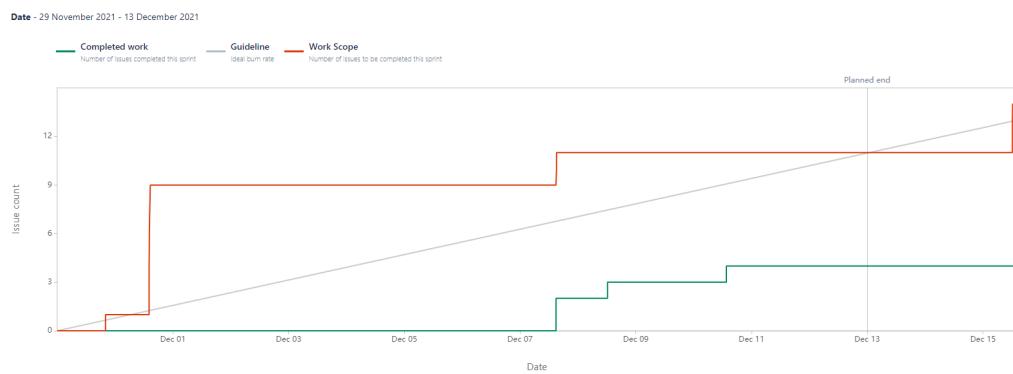


Figure 35: BurnUp graf šprintu č.15

Čo sme spravili dobre?

- Začali sme sa pripravovať na finalizáciu TP
- Dohodli sme sa na rozdelení konferencií
- Dohodli sme testovanie produktu
- Kontinuálne zlepšovanie UX

Čo nám robilo problémy?

- Pokuľhávame najmä na fronte grafových neurónov z dôvodu nedostatku času
- Akosi rýchlo prišiel koniec semestra - treba napísat miňník

Návrhy na zlepšenie

- Vynájsť stroj času
- Predĺžiť trvanie dňa na 30h

Prílohy

A Metodika zabezpečovania kvality kódu

Cieľom tejto metodiky je zadefinovanie si sady pravidiel o písaní kódu v Pythone. Pre tieto účely môžeme použiť štandardy PEP ako aj niekoľko vlastných pravidiel. Pri písaní kódu sa následne vieme riadiť uvedenými pravidlami, ktoré sa ďalej kontrolujú pri posúdení kódu.

Pravidlá písania kódu

Pre posúdenie dodržania štandardov a zachovanie kvality Python kódu vieme použiť nástroj na statickú analýzu kódu PyLint. Štandard, ktorý PyLint požaduje aby sa dodržiaval, je *PEP-8*. Okrem *PEP-8* sa však vyžaduje dodržiavanie ďalších pravidiel uvedených v tejto metodike. V prípade, ak nejaké pravidlo nie je uvedené v tomto dokumente, ale definuje ho štandard, je vysoko odporúčané sa riadiť daným štandardom. Táto metodika slúži primárne ako návod, avšak špecifické prípady si môžu vyžadovať výnimky (ich opodstatnenie sa posúdi počas code review).

Dĺžka riadku

- Maximálna dĺžka riadku je podľa *PEP-8* 79 znakov. PyLint tento limit rozšíril na 100 znakov, čo považujeme za vhodnejší limit, a preto ho použijeme aj v našej metodike.
- V prípade prekročeného limitu dĺžky riadku treba dodržať zásady odsadenia.

Odsadenie, prázdne riadky a medzery

- Na konci riadku by nemali byť zbytočné medzery. Taktiež je potrebné sa vyvarovať zbytočným znakom medzery vnútri kódu (napr. medzi názvom funkcie a zátvorkou s argumentami), obsiahlo túto časť dokumentuje *PEP-8*.
- Medzi funkciami treba mať 2 riadky voľné. Na konci súboru má byť práve jeden prázdnny riadok. Logické skupiny importov modulov majú

byť oddelené prázdnym riadkom (viď. nižšie). Ak za importami nasleduje definícia globálnych premenných, prípadne nastavenia (napr. logovanie), pridáva sa za importy jeden prázdný riadok, v opačnom prípade dva prázdne riadky.

Pomenovávanie premenných, funkcií, tried atď.

- Všetky vlastné pomenovania musia byť písané v anglickom jazyku.
- V prípade premenných treba dodržiavať snake_case. Niektoré premenné môžu pravidlá písania daným štýlom narúšať (napr. df - dataframe). Takéto prípady patria medzi výnimky a sú povolené, musí však ísť o všeobecne známe a používané názvy.
- Pre moduly (Python súbory) a balíky je vhodné používať malé písmená. Ideálne by mali byť tieto názvy čo najkratšie (jednoslovné). Ak je to nutné, je možné používať snake_case. Pridaním znaku _ na rozdelenie slov vieme dosiahnuť lepšiu čitateľnosť a priamejšie pomenovanie.
- Názvy tried by mali používať konvenciu CapWords.
- Používané názvy by mali výstižne pomenovať ich zmysel a funkcionalitu. Je lepšie názov rozpísat, než používať neznámu skratku, ktorej význam si nikto nedomyslí.

Nápoveda dátového typu

- Cieľom tohto pravidla je využitie anotácie dátových typov, čím sa Python kód stane prístupnejším pre statickú analýzu kódu a refactoring. Aplikujú sa pravidlá smerníc PEP 484 a PEP 3107.
- Nápovedy je nutné definovať spôsobom:
 - parametre: def foo(a: expression, b: expression = 5): .
 - návratové hodnoty: def sum() -> expression: .

Import modulov

- Importy musia byť vždy vložené na začiatok modulu, ale až po komentároch pre dokumentáciu.

- Poznáme tri druhy importov. V module musia byť importy rovnakého druhu zoskupené spolu a oddelené prázdnym riadkom. Konkrétnie ide o tieto typy importov (treba dodržať aj poradie typov importov):
 - importy zo štandardnej knižnice
 - importy zo súvisiacich knižníc tretích strán
 - importy z lokálneho prostredia
- Nepoužívané importy treba vymazat.

Refaktorинг

- V prípade, ak sa opakuje niektorá funkcia v kóde, je potrebné sa zamyslieť nad možnosťou refaktorovania kódu. Medzi takéto možnosti môže patriť vypichnutie opakujúcej sa časti kódu a jej oddelenie v samostatnej funkcií či rozdelenie komplikovanej funkcie na menšie časti.
- Pomocou refaktoringu môžeme docieliť nielen lepšiu čitateľnosť, ale aj znova-použiteľnosť. V snahe docieliť modularizáciu vieme kód rozpísat do funkcií či tried.
- Treba dodať, že nie je nutné za každú cenu refaktorovať. Dôležité je, aby jednotlivé časti kódu (funkcie, triedy) dávali zmysel a nevenovali sa príliš veľkému počtu funkcionálít.

Dokumentácia v kóde

- Štandardy *PEP-8* a *PEP-257* opisujú pravidlá písania dokumentácie v Python kóde.
- Pre dokumentáciu sa používajú dokumentačné reťazce (docstrings), majúce tvar dvojitých úvodzoviek.
- Napísanie dokumentácie je potrebné pre moduly, funkcie, triedy a ich metódy.

Ostatné pravidlá

- Textové reťazce budú definované jednotne, a to s použitím jednoduchých úvodzoviek v prípade identifikátorov a dict kľúčov a dvojitych úvodzoviek v ostatných prípadoch. V prípade, ak string obsahuje dvojitú úvodzovku, treba použiť na definovanie reťazca jednoduché úvodzovky a opačne, aby sa zabránilo zbytočnému používaniu escape znakov.
- Spustiteľný súbor musí mať definovanú časť
`if __name__ == "__main__":`
- Formátovanie v logovaní nemusí používať lazy formatting (%), aj keď ho odporúča PyLint. Lazy formatting sa síce v *PEP-282* spomína, no nie je uvedený ako štandard.
- V prípade, ak chce programátor použiť v jazyku Python konštrukciu podobnú rozhraniu pre triedu, mal by vytvoriť abstraktnú triedu podľa smernice *PEP-3119*.
- Čokoľvek nepoužité (premenná, funkcia, trieda, modul ...) by malo byť vymazané, prípadne treba v komentári vysvetliť, prečo daný kód nevymazať.

B Metodika testovania

Súčasťou vyvíjaného produktu je aj jeho testovanie a validácia. V tejto metodike možno nájsť konkrétnie informácie týkajúce sa implementovania testov do projektu.

Jednotkové testy

Prostredníctvom jednotkových testov vieme zistiť správnosť fungovania určitej ucelenej časti kódu (funkcia, trieda, metóda v triede ...). Jednotkové testy by mal prednostne písat autor danej časti kódu, prípadne sa napíšu dodatočne v procese QA podľa potreby. Zoznam pravidiel pri písaní testov:

- Python: testy sa musia nachádzať v balíku tests.
- Python: konvencia názvoslovia testovacích modulov postupuje podľa priyatých pravidiel písania kódu (teda *snake_case*), každý názov testovacieho modulu ale musí začínať na *test_*, teda napr. *test_your_module.py*.
- Python: jednotkové testy sa vykonávajú pomocou knižnice *pytest*.
- Princíp AAA (arrange, act, assert). Ak je to možné, je odporúčané dodržiavať tento princíp. Najskôr si pripravíme všetko, čo k vykonaniu testu potrebujeme (arrange). To môže byť vytvorenie inštancie objektu, naplnenie premenných objektu nejakými hodnotami a podobne. Následne vykonáme funkcionality, ktorú chceme testovať (act), čo môže byť napr. zavolanie metódy. Nakoniec chceme porovnať očakávaný výsledok s reálnym výsledkom (assert), čím si overujeme hypotézu daného testu.
- Jeden test by mal testovať práve jednu hypotézu. Test sa tak musí venovať jednej konkrétnej funkcionalite, aby sa zabránilo nedorozumeniam v interpretácii výsledkov. Odporúčaný je jeden assert na test, nie je to však nutné, pokial sa asserts venujú rovnakej hypotéze.
- Každý test by mal mať svoju krátku dokumentáciu, aby bolo jasné, čo sa testuje a aký výsledok sa očakáva.

C Metodika verziovania kódu

Na kolaboratívnu prácu na projekte sa používa nástroj na verziovanie kódu Github. V rámci organizačnej štruktúry máme pridelený priestor organizácie STU-FIIT-Team-20-2021. V danej organizácii si vytvoríme súkromné repozitáre, ktoré odzrkadľujú jednotlivé logické a organizačné celky projektu. Súčasťou projektu sú tieto Github repozitáre:

- website - súbory súvisiace so statickou verziou webového sídla tímu
- data-pipeline - kód zaoberajúci sa získavaním a čistením dát
- phototox-webapp - webová aplikácia, s ktorou bude interagovať používateľ
- phototox-analysis - venuje sa analýze získaných dát
- chem-descriptor-evaluator - vyhodnocovanie chemických deskriptorov zo SMILES kódov

Súčasťou repozitárov sú vetvy, ktoré umožňujú bezpečné vyvíjanie nových funkcionálít či opravovanie chýb. Hlavnú vetvu pomenúvame *master*, prípadne *main*. Vedľajšie vetvy sa následne pomenúvajú podľa ich významu a dôvodu vzniku. Dôležité je udržiavanie správnosti *master* vetvy - nedokončené či neoverené funkcionality je vhodné najskôr vyriešiť v samostatnej vetve až potom urobiť zlúčenie vetiev.

Špeciálne vettové pravidlá nájdeme v repozitáre phototox-webapp, kde *Main* vetva je chránená proti Pushovaniu kódu, a tím vie do tejto hlavnej vetvy pridávať zmeny len pull-requestom z *Test* vetvy, a tento pull-request sa schváli len za určitých podmienok, čo je bližšie popísané v Appendix D.

D CICD Pipeline

Spojenie metodiky verziovania, metodika testovania a automatický deployment spolu tvoria nás CICD pipeline. V rámci nášho repozitára pre phototox-webapp, čo je hlavné sýdlo našej webovej aplikácie ako kombinácia všetkých jej súčastí z predošlých repozitárov, obsahuje dve hlavné vetvy a to *Main* a *Test*.

Do vetvy main sa nedá priamo prispevať pushom, iba pull-requestom z testu. Ked nastane pull-request v Testo branchy do Main, tak sa spustia workflowy v GitHub Actions, ktoré obsahujú automatický deployment repozitáru na Linuxový stroj, kde sa vykonajú nami definované Django testy.

Zároveň sa vykoná kontrola bezpečnosti kódu, a kontrolujú sa aj verzie použitých packagov a libraries, ak by sme náhodou používali library, ktorá v sebe má nejakú security vulnerability, na čo nás GitHub upozorní a vie automaticky takúto skutočnosť napraviť.

Ak zbehnú testy úspešne, pull-request sa automaticky potvrdí a zmeny sa zapracujú do Main vetvy. Main vetva nemôže nikdy predbehnúť Test vetvu, čiže spory môžu nastáť len v Test vetve kde sa môžu riesiť.

Na stroji máme deployment kód, na ktorý sa dá poslať request aby nastal automatický deployment. Tento PHP skript overí z akej IP adresy a s akým obsahom prišiel request, a ak sa zhoduje s povoleným vzorom, tak následne si stiahne najnovšiu verziu webaplikácie z repozitáru, a zapracuje nové zmeny v databáze.

Tento request pochádza z GitHubu ako WebHook v momente ked sa updatne Main vetva v repozitáry. Zároveň treba poznamenať, že pri vývoji používame lokálne nastavenia Djanga pre testovanie, a zároveň sú všetky zmeny databázy najprv pre test vykonávané do lokálnej databázy SQLite. Tieto zmeny sú sledované a následne aplikované na produkčnú databázu. Výhodou je, že ak nastali viaceré konfliktné zmeny databázy z rôznych úloh, kvôli takémuto sledovaniu zmien v lokálnej databáze vieme urobiť na zmeny merge a často automaticky rezolvovať konflikty, alebo odhaliť kritický konflikt ešte vo vetve Test.

Zjednodušený proces CICD je znázornený v Petriho sieti nižšie. CICD Pipeline nám dovoluje bezstarostne vykonávať zmeny v kóde a takmer hned vidieť výsledky na live stránke bez zbytočných starostí, a zachytávať problémy než sa dostanú do produkcie.

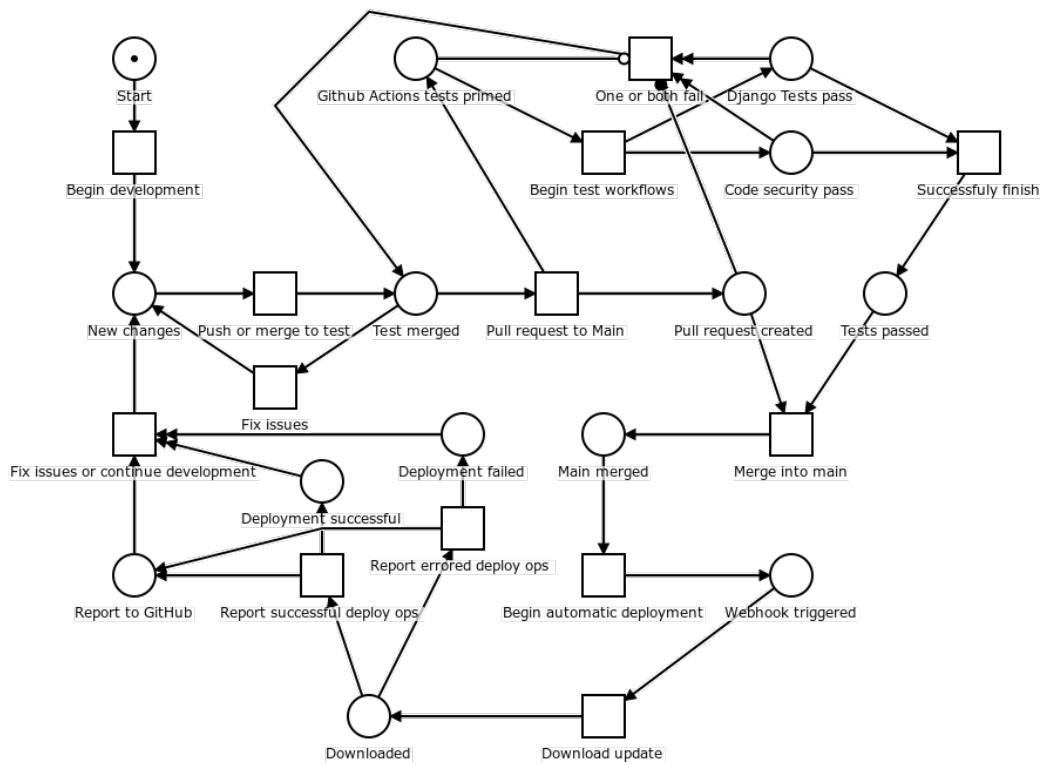


Figure 36: Znázormenie CICD procesu Petriho sietou.

E Metodika správy úloh

Pre správu a sledovanie úloh v našom projekte používame nástroj Jira od spoločnosti Atlassian. Umožnuje nám jednoducho evidovať úlohy, ktoré sú potrebné spraviť v danom špriente a kto je za ne zodpovedný. Tento nástroj poskytuje rozsiahlu funkciaľitu a značne nám zjednoduší udržiavanie prehľadu nad vykonávanou prácou.

Funkciaľitu nástroja sme si upravili tak, aby vyhovovala naším potrebám a vytvorili sme nasledujúce rámce používania, ktoré zabezpečujú čo najplynulejšiu spoluprácu a umožňujú každému členovi tímu zostať v obraze.

V rámci letného semestra sme upravili náš workflow na 1 týždňové šprinty, čo nám umožnilo pracovať dynamickejšie a lepšie odhadovať prácu, ktorú zvládneme spraviť. V zimnom semestri sme mali problémy s odhadovaním vykonateľnosťou niektorých úloch, ktoré boli ovplyvnené externými faktormi, ktoré sme nevedeli predpovedať.

Zaviedli sme aj systém story points, založený na pridelení najľahšej úlohe týždňa 1 bod a zvyšným úloham body podla ich relatívnej náročnosti. Tento systém nám umožnil presnejšie odhadnúť naše kapacity na daný šprint a rozdeliť prácu rovnomernejšie medzi členov tímu.

Zaviedli sme aj systém "Definition of Done", ktorý presne špecifikoval ciele pre každú úlohu, aby sa nenastali problémy keď členovia tímu očakávali z tej istej úlohy rôzne výstupy.

E.1 Workflow

Predvolený workflow v Jira je pre naše potreby príliš jednoduchý. Pozostáva iba z 3 základných častí TO DO → IN PROGRESS → DONE. V našej metodike používame ešte dva kroky navyše.

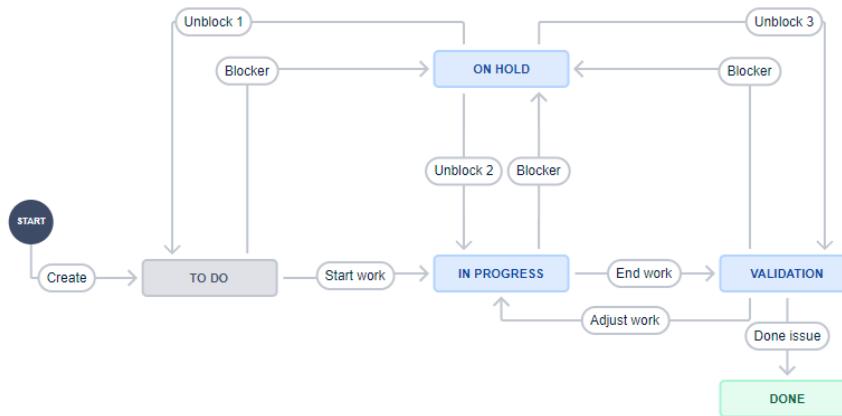


Figure 37: Workflow v Jira

- **TO DO** - Úloha je vytvorená a pridelená členovi tímu, ale ten na nej ešte nezačal pracovať.
- **IN PROGRESS** - Práca na úlohe prebieha.
- **ON HOLD** - Práca na úlohe bola zablokovaná vzniknutým problémom. Zväčša znamená čakanie na dokončenie inej úlohy.
- **VALIDATION** - Po ukončení práce na úlohe musí prejsť validáciou, ktorej cieľom je overiť či vykonaná práca skutočne splnila požiadavky.
- **DONE** - Uzavretá úloha.

E.2 Typy úloh

Jira poskytuje rôzne úrovne granularity úloh a umožňuje začleňovať úlohy do logických skupín. Jedna z možností sú napríklad Story points, užitočné pre Agile vývoj. Avšak prostredie Jiry nie je pre využitie Story points intuitívne, preto sa spoliehame primárne na Tasks, Sub-Tasks a Epics. Do Epics začleňujeme všetky úlohy a to podľa nasledujúceho klúča:

- **Paperwork** - Administratívne úlohy ako záznamy so stretnutí, retrospektívy šprintov a písanie milníku
- **Data engineering** - Získavanie nový dát zo zdrojov, vývoj všetkých častí data pipeline
- **Modelling** - Zavádzanie systému pre automatické iteratívne vylepšovanie modelu
- **MLOps** - Experimentálny vývoj modelov, pretrénovanie existujúcich modelov na nových dátach a podobne
- **Web** - Úlohy týkajúce sa ako údržby tímového webu, tak aj vývoju webovej aplikácie

E.3 Role

Rola	Zodpovednosti
Majiteľ	Vytváranie úloh Písanie popisu úlohy Určí kto je za úlohu zodpovedný Validuje úlohu Môže aktualizovať status úlohy
Zodpovedný	Zodpovedný za prácu na úlohe Komunikuje s Majiteľom a Podporou Môže delegovať časť práce pre Podporu Aktualizuje stav úlohy
Podpora	Podporuje Zodpovedného programovaním, diskusiou, zosúladením s inou úlohou, ... Neaktualizuje status úlohy Komunikuje hlavne so Zodpovedným Nie je zodpovedný za úlohu

E.4 Komunikácia

Aktualizácia	Obsah
Povinná	Pred každým stretnutím Pri presune úlohy do stavu ON HOLD - treba špecifikovať čo úlohu blokuje Neakceptovanie úlohy pri validácii Čo najdetailnejšie popísať čo nebolo splnené.
Dobrovoľná	Inkrementálne aktualizácie keď je úloha v stave IN PROGRESS Možnosť upozorniť (@) člena tímu pri zmene stavu Kedykoľvek to člen tímu uzná za vhodné

F Metodika dátovej vedy

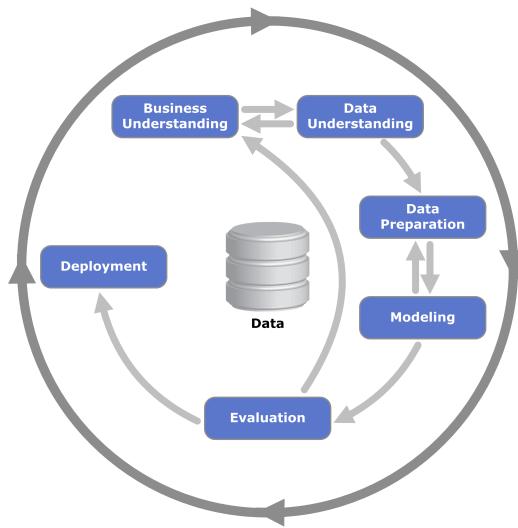


Figure 38: CRISP-DM

Naša práca z veľkej časti spočíva v dátovej vede a strojovom učení. Táto podskupina softvérového inžinierstva má kvôli podstate svojej činnosti špecifické požiadavky na pracovný postup. V našom tíme sa teda okrem Agile metodológie spoliehame z veľkej časti aj na CRISP-DM (Cross Industry Standard Process for Data Mining).

CRISP-DM popisuje prirodzený životný cyklus, ktorý sa vykryštalizuje vo väčšine projektov dátovej vedy. S naším prehľbujúcim sa porozumením problematike fototoxicity sa prehľbuje aj pochopenie dôležitosti rôznych deskriptorov, ktorými ju môžeme predikovať. To nám umožňuje upravovať spôsob prípravy dát a trénovania modelu tak, aby sme dosahovali čoraz lepšie výsledky. V našej webovej aplikácii nám to umožňuje stále používať najlepší dostupný model, ale zároveň sme pripravený na zmeny a implementáciu vylepšení.

G Používané chemické deskriptory

Na základe štruktúry molekúl látok a ich chemických vlastností, sme identifikovali alebo vypočítali rôzne užitočné deskriptory, ktoré by mohli byť korelované s fototoxicitou látok. Následne sme nad týmito deskriptormi trénovali modely strojového učenia, a analyzovali užitočnosť a teda dôležitosť rôznych deskriptorov pre klasifikáciu látok medzi fototoxické a nefototoxické. Samotná dôležitosť deskriptorov na základe natrénovaného modelu je viedieť vo Figure 39.

Deskriptor	Popis deskriptoru
cycles	Počet karbocyklov v molekule
atom_valence	Suma valencie atómov v molekule
gesteiger (rôzne)	Rôzne štatistické hodnoty odvodené z Gesteigerových nábojov atómov, napr. suma pozitívnych a negatívnych nábojov, ich priemery, a pod.
hydrogen_count	Počet vodíkov v molekule
csp3	Pomer sp3 hybridizovaných uhlíkov ku všetkým uhlíkov
aromatic_heavy_atoms	Počet ťažkých atómov v aromatických cykloch
balban_j	Topologický index komplexity molekuly
hallkier	Pomer kovalentných polomerov pre nehybridizované atómy uhlíka
labute_asa	Obsah prístupnej plochy molekuly pre solvent
h_donors/h_acceptors	Vodíkové donory a akceptory
carbon_count	Počet uhlíkov
tpsa	Obsah polárnej povrchovej plochy molekuly
druglikeness	Počet violácií deskriptorov, ktoré identifikujú molekulu ako farmaceutický cieľ
vsa_logP_N	Rôzne atribúty vypočítanej lipofolity molekúl
cycle_type_counts	Log16 hodnôt namapovaných počtov cyklov rôznych veľkostí na jednotky hexadecimálnej sústavy s ofsetom rovným veľkosti karbocyklu

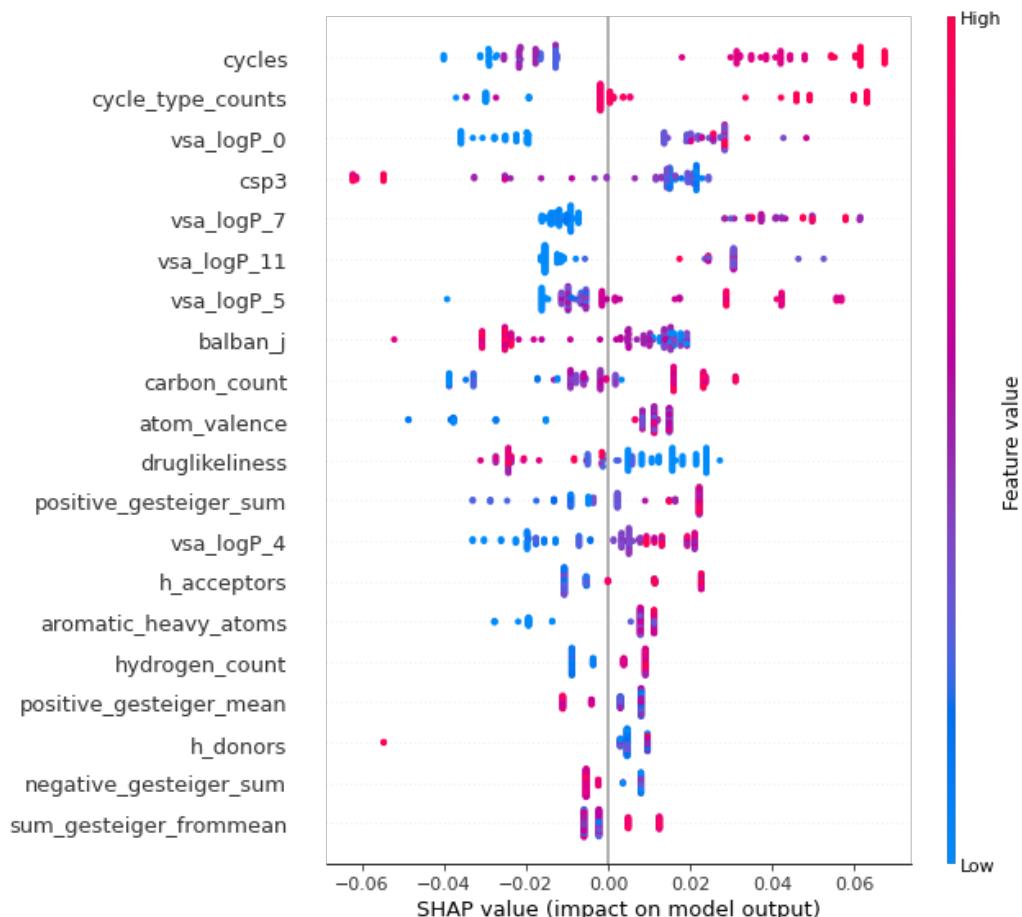


Figure 39: SHAP dôležitosť deskriptorov. Os X znázorňuje vplyv premennej na rozhodnutie. Záporné hodnoty na tejto osi predstavujú naklonenie ku klasifikovaniu látky ako 'nefototoxiccká', kladné hodnoty pre modelu zvyšujú šancu klasifikovať ju ako 'fototoxiccká'. Os Y predstavuje 20 najdôležitejších deskriptorov zoradených zostupne. Farba jednotlivých bodov v grafe predstavuje hodnotu deskriptoru. Teda pre deskriptrov 'cycles' vidíme, že nízke hodnoty tohto deskriptoru považuje model za indikátory nefototoxickej látky a vysoké hodnoty považuje za indikátor fototoxickej látky.

H Metodika komunikácie

Našim primárnym komunikačným kanálom je súkromný discord server. V rámci neho sa nachádzajú tri kanály, a to *general*, *paperwork* a *doc*. Prvý spomínaný slúži na všeobecnú komunikáciu, v druhom sa zhromažďujú materiály k písomným prácам, ktoré vznikajú v priebehu semestra ako napríklad burndown grafy. Posledný spomínaný kanál obsahuje predovšetkým odkazy na užitočné vedecké články vzťahujúce sa k riešenej problematike. Možno tu nájsť aj verejné databázy chemických látok (napr. PubChem, Swiss, ...), alebo OECD materiály. Konverzácia je prevažne štruktúrovaná pomocou vlákien.

Záložným variantom pre prípad núdze je telefonický kontakt, ktorý každý z nás na začiatku poskytol. Doteraz sme ho však nemuseli využiť.

Ďalšie kanály slúžia na komunikáciu s vedúcou nášho tímu. Na tento účel sme zriadili spoločný gmailový účet, cez ktorý sú zdieľané články, alebo komplexnejšie informácie. Pokiaľ je treba rýchlo čokoľvek odkomunikovať, je na to použitá WhatsApp skupina.

Kedže sa od polovice novembra zhoršila epidemiologická situácia v Bratislave natoľko, že sme prešli do dištančnej výučby, nebolo nadalej možné sa prezenčne schádzať. Pri tímových stretnutiach sme nadalej pokračovali na platforme Discord, pri stretnutiach s vedúcimi využívame GoogleMeet.

PRÍLOHA I METODIKA UKLADANIA DOKUMENTOV A INFORMÁCIÍ-1

I Metodika ukladania dokumentov a informácií

Spolu s nástrojom Jira od spoločnosti Atlassian používame aj Confluence. Primárne slúži na ukladanie interných dát a dokumentov. Pod dátami rozumieme súbory obsahujúce fototoxicitu molekúl, vypočítané deskriptory, alebo extrahované descriptory.

Dokumenty sú trojakého druhu. V prvom z nich sú popísané metodológie, a ďalej potom popis architektúry spolu s grafmi. Najdôležitejšou časťou sú súbory so získanými informáciami súvisiacimi s fototoxicitou, ako je chemická podstata, alebo spôsob merania.

J Export úloh

Šprint č.1

Issue Type	Issue key	Issue id	Summary	Assignee	Reporter	Status	Resolution	Created	Updated
Task	PP-35	10034	Feature importances - SHAP	Matej Halinkovic	Matej Halinkovic	Done	Done	13/10/2021 15:44	14/10/2021 18:08
Task	PP-19	10018	Extract predictions from SwissTarget	Matej Halinkovic	Matej Halinkovic	Done	Done	06/10/2021 14:46	14/10/2021 18:08
Task	PP-4	10003	Setup Atlassian products	Tibor Sloboda	Matej Halinkovic	Done	Done	06/10/2021 13:27	06/10/2021 14:46
Task	PP-3	10002	Extract Data from Pubchem	Matej Halinkovic	Matej Halinkovic	Done	Done	06/10/2021 13:26	14/10/2021 18:08
Task	PP-2	10001	Create report template	Katerina Muskova	Matej Halinkovic	Done	Done	06/10/2021 13:26	14/10/2021 18:08
Task	PP-1	10000	Setup website	Patrik	Matej Halinkovic	Done	Done	06/10/2021 13:25	14/10/2021 18:10

PRÍLOHA J EXPORT ÚLOH

J-2

Šprint č.2

Issue Type	Issue key	Issue id	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-64	10063	Jira methodology design	Matej Halinkovič	Matej Halinkovič	Done	Done	31/10/2021 12:16	31/10/2021 12:16
Task	PP-56	10055	Cleaner clustering	Kateřina Mušková	Matej Halinkovič	Done	Done	27/10/2021 14:31	31/10/2021 12:16
Task	PP-51	10050	Populate CSV with Chem. properties (pipeline step)	Tibor Sloboda	Tibor Sloboda	Done	Done	21/10/2021 21:11	26/10/2021 12:29
Task	PP-50	10049	Data pipeline architecture	Matej Halinkovič	Matej Halinkovič	Done	Done	20/10/2021 17:46	27/10/2021 14:52
Task	PP-47	10046	CSV Merger	Tibor Sloboda	Matej Halinkovič	Done	Done	19/10/2021 15:14	26/10/2021 12:29
Task	PP-46	10045	Identify cyclic and acyclic compounds	Tibor Sloboda	Matej Halinkovič	Done	Done	19/10/2021 15:14	21/10/2021 20:50
Task	PP-40	10039	Ensemble learning exploration	Matej Halinkovič	Matej Halinkovič	Done	Done	13/10/2021 22:35	27/10/2021 14:32
Task	PP-39	10038	Deeper EDA - properly identify outliers, distributions, etc.	Kateřina Mušková	Matej Halinkovič	Done	Done	13/10/2021 22:34	27/10/2021 14:52
Task	PP-38	10037	Explore different architectures: Logit, SVM, ...	Matej Halinkovič	Matej Halinkovič	Done	Done	13/10/2021 22:33	27/10/2021 12:40
Task	PP-37	10036	Iteratively improve website	Patrik	Tibor Sloboda	Done	Done	13/10/2021 18:37	26/10/2021 14:30
Task	PP-32	10031	Re-train RandomForest with new data	Matej Halinkovič	Tibor Sloboda	Done	Done	13/10/2021 15:39	27/10/2021 12:40

PRÍLOHA J EXPORT ÚLOH

J-3

Task	PP-29	10028	Meeting report 4	Jakub Maruniak	Tibor Sloboda	Done	Done	13/10/2021 15:37	27/10/2021 13:15
Task	PP-27	10026	Meeting report 3	Matej Halinkovič	Tibor Sloboda	Done	Done	13/10/2021 15:35	27/10/2021 12:40
Task	PP-21	10020	Obtain additional data	Jakub Maruniak	Matej Halinkovič	Done	Done	13/10/2021 15:23	19/10/2021 15:10
Task	PP-20	10019	TP Cup application	Jakub Knánik	Matej Halinkovič	Done	Done	10/10/2021 10:10	31/10/2021 12:15
Task	PP-9	10008	Meeting report 2	Jakub Maruniak	Tibor Sloboda	Done	Done	06/10/2021 14:06	19/10/2021 15:10

PRÍLOHA J EXPORT ÚLOH

J-4

Šprint č.3

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-96	Create doxygen	Kateřina Mušková	Tibor Sloboda	Done	Done	13/Nov/21 6:20 PM	16/Nov/21 1:24 PM
Task	PP-54	Data extraction from newspapers	Jakub Maruniak	Matej Halinkovič	Done	Done	27/Oct/21 2:30 PM	16/Nov/21 4:28 PM
Task	PP-57	Data pipeline assembling	Tibor Sloboda	Matej Halinkovič	Done	Done	27/Oct/21 2:31 PM	10/Nov/21 2:47 PM
Task	PP-58	Data pipeline code review	Jakub Knánik	Matej Halinkovič	Done	Done	27/Oct/21 2:32 PM	16/Nov/21 1:11 PM
Task	PP-65	Expand calculated descriptors	Tibor Sloboda	Matej Halinkovič	Done	Done	02/Nov/21 4:00 PM	07/Nov/21 2:10 PM
Task	PP-93	Experiment with unsupervised approach	Matej Halinkovič	Matej Halinkovič	Done	Done	10/Nov/21 3:20 PM	16/Nov/21 1:20 PM
Task	PP-67	Find conditions of the Muegge filter	Jakub Knánik	Tibor Sloboda	Done	Done	03/Nov/21 5:12 PM	03/Nov/21 5:36 PM
Task	PP-68	Japan data extraction	Matej Halinkovič	Matej Halinkovič	Done	Done	09/Nov/21 3:29 PM	16/Nov/21 1:19 PM
Task	PP-60	Meeting report 4	Matej Halinkovič	Matej Halinkovič	Done	Done	27/Oct/21 2:43 PM	02/Nov/21 4:01 PM
Task	PP-61	Meeting report 5	Jakub Knánik	Matej Halinkovič	Done	Done	27/Oct/21 2:43 PM	03/Nov/21 2:54 PM
Task	PP-76	Meeting report 6	Tibor Sloboda	Matej Halinkovič	Done	Done	09/Nov/21 3:53 PM	15/Nov/21 3:01 PM
Task	PP-66	Modify cleaner to create masks instead of deletion	Kateřina Mušková	Tibor Sloboda	Done	Done	03/Nov/21 2:23 PM	10/Nov/21 1:28 PM
Task	PP-63	Modify cleaner to fit data pipeline	Kateřina Mušková	Tibor Sloboda	Done	Done	30/Oct/21 11:51 AM	03/Nov/21 2:25 PM
Task	PP-97	Replace seaborn in cleaner by alternativy	Kateřina Mušková	Tibor Sloboda	Done	Done	14/Nov/21 10:27 AM	14/Nov/21 1:54 PM

PRÍLOHA J EXPORT ÚLOH

J-5

Task	PP-53	Setup flask website prototype	Matej Halinkovič	Matej Halinkovič	Done	Done	27/Oct/21 2:29 PM	10/Nov/21 1:02 PM
Task	PP-52	Sprint 1 retrospective	Jakub Maruniak	Matej Halinkovič	Done	Done	27/Oct/21 1:01 PM	02/Nov/21 3:50 PM
Task	PP-62	Sprint 2 retrospective	Matej Halinkovič	Matej Halinkovič	Done	Done	27/Oct/21 2:43 PM	10/Nov/21 1:21 PM
Task	PP-100	Updating website	Patrik	Patrik	Done	Done	16/Nov/21 2:54 PM	16/Nov/21 2:55 PM
Task	PP-88	Validate Data Cleaner on calculated descriptors	Kateřina Mušková	Matej Halinkovič	Done	Done	10/Nov/21 1:28 PM	13/Nov/21 5:40 PM
Task	PP-55	Viability of calculated descriptors	Matej Halinkovič	Matej Halinkovič	Done	Done	27/Oct/21 2:31 PM	07/Nov/21 2:09 PM

PRÍLOHA J EXPORT ÚLOH

J-6

Šprint č.4

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-101	Retrain models with Spielmann data	Matej Halinkovič	Matej Halinkovič	Done	Done	16/11/2021 16:29	27/11/2021 10:13
Task	PP-99	Create diagram for modelling experiments	Matej Halinkovič	Matej Halinkovič	Done	Done	16/11/2021 14:24	24/11/2021 13:54
Task	PP-98	Sprint 3 Retrospective	Patrik	Matej Halinkovič	Done	Done	16/11/2021 13:28	24/11/2021 17:25
Task	PP-92	Reformat existing documents to Latex	Kateřina Mušková	Kateřina Mušková	Done	Done	10/11/2021 14:37	23/11/2021 10:38
Task	PP-90	Doxigen generation	Tibor Sloboda	Kateřina Mušková	Done	Done	10/11/2021 14:31	29/11/2021 19:57
Task	PP-78	Design initial web app architecture for the Milestone	Tibor Sloboda	Matej Halinkovič	Done	Done	09/11/2021 16:01	29/11/2021 20:01
Story	PP-75	Milestone I.	Matej Halinkovič	Matej Halinkovič	Done	Done	09/11/2021 15:51	27/11/2021 10:13
Story	PP-74	Web app architecture	Tibor Sloboda	Matej Halinkovič	Done	Done	09/11/2021 15:38	29/11/2021 20:01
Task	PP-59	Milestone documentation	Matej Halinkovič	Matej Halinkovič	Done	Done	27/10/2021 14:37	27/11/2021 10:13

Šprint č.5

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-116	Sprint retrospective 5	Matej Halinkovič	Matej Halinkovič	Done	Done	15/12/2021 12:13	15/12/2021 13:32
Task	PP-115	Meeting report 11	Jakub Maruniak	Matej Halinkovič	Done	Done	15/12/2021 12:13	15/12/2021 13:32
Task	PP-114	Milestone II.	Matej Halinkovič	Matej Halinkovič	Done	Done	08/12/2021 16:39	15/12/2021 13:32
Task	PP-113	Meeting report 10	Matej Halinkovič	Matej Halinkovič	Done	Done	07/12/2021 15:07	15/12/2021 13:30
Task	PP-112	Explore Django JavaScript integration	Patrik	Matej Halinkovič	Done	Done	07/12/2021 15:07	15/12/2021 13:31
Task	PP-110	Basic UX design	Tibor Sloboda	Matej Halinkovič	Done	Done	30/11/2021 14:37	15/12/2021 13:30
Task	PP-109	Create database prototype for webapp	Katerína Mušková	Matej Halinkovič	Done	Done	30/11/2021 14:30	10/12/2021 13:34
Task	PP-108	Analyze ORCID authentication methods	Patrik	Matej Halinkovič	Done	Done	30/11/2021 14:21	15/12/2021 13:32
Task	PP-107	Port Webapp prototype to Django	Matej Halinkovič	Matej Halinkovič	Done	Done	30/11/2021 14:14	07/12/2021 14:55
Task	PP-106	Data pipe testing design	Jakub Knánik	Matej Halinkovič	Done	Done	30/11/2021 14:12	15/12/2021 13:32
Task	PP-104	Retrain model with aminoacids	Matej Halinkovič	Matej Halinkovič	Done	Done	30/11/2021 14:08	07/12/2021 14:54
Task	PP-103	Sprint retrospective 4	Matej Halinkovič	Matej Halinkovič	Done	Done	30/11/2021 14:07	15/12/2021 13:30
Task	PP-102	Meeting report 9	Patrik	Matej Halinkovič	Done	Done	30/11/2021 14:07	08/12/2021 12:16

Šprint č.6

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-126	Convert extracted HTML to Django Templates	Patrik	Matej Halinkovič	Done	Done	18/02/2022 19:40	24/02/2022 11:31
Task	PP-125	Alvadesc feature cleaning	Tibor Sloboda	Matej Halinkovič	Done	Done	17/02/2022 11:39	19/02/2022 21:08
Task	PP-124	Alvadesc molecule merge	Tibor Sloboda	Matej Halinkovič	Done	Done	17/02/2022 10:55	19/02/2022 21:08
Task	PP-122	Implement IP adress table	Kateřina Mušková	Matej Halinkovič	Done	Done	17/02/2022 10:51	24/02/2022 09:23
Task	PP-121	Migrate DB to Postgres	Kateřina Mušková	Matej Halinkovič	Done	Done	17/02/2022 10:50	24/02/2022 09:23
Task	PP-120	Marvin JS API implementation	Jakub Knánik	Matej Halinkovič	Done	Done	17/02/2022 10:50	24/02/2022 11:31
Task	PP-119	Alvadesc feature importances	Matej Halinkovič	Matej Halinkovič	Done	Done	17/02/2022 10:46	21/02/2022 17:20
Task	PP-118	Analyze extracted HTML from Axure	Patrik	Matej Halinkovič	Done	Done	17/02/2022 10:44	18/02/2022 19:36
Task	PP-117	Meeting report, 1st week of semester	Matej Halinkovič	Matej Halinkovič	Done	Done	17/02/2022 10:43	21/02/2022 17:20

Šprint č.7

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-131	Modelling MLP experiment	Matej Halinkovič	Matej Halinkovič	Done	Done	23/02/2022 13:37	02/03/2022 12:50
Task	PP-133	Meeting report 2	Jakub Knánik	Matej Halinkovič	Done	Done	24/02/2022 11:32	02/03/2022 12:50
Task	PP-135	Prediction descriptors output	Matej Halinkovič	Matej Halinkovič	Done	Done	24/02/2022 11:34	02/03/2022 12:24
Task	PP-137	Django user input validation research	Jakub Knánik	Matej Halinkovič	Done	Done	24/02/2022 11:40	26/02/2022 19:25
Task	PP-138	Consult Juraj Petrik about web hosting	Tibor Sloboda	Matej Halinkovič	Done	Done	24/02/2022 11:45	02/03/2022 13:02
Task	PP-141	Prediction corrections frontend	Patrik	Matej Halinkovič	Done	Done	24/02/2022 11:50	03/03/2022 10:06
Task	PP-142	Sprint retrospective 1	Matej Halinkovič	Matej Halinkovič	Done	Done	26/02/2022 18:04	02/03/2022 12:24
Task	PP-143	Create login/register pages	Jakub Knánik	Matej Halinkovič	Done	Done	26/02/2022 19:02	03/03/2022 10:09
Task	PP-145	Setup SMTP server / Gmail account for verification emails	Kateřina Mušková	Matej Halinkovič	Done	Done	26/02/2022 19:10	02/03/2022 12:50

PRÍLOHA J EXPORT ÚLOH

J-10

Šprint č.8

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-157	Implement URL field for submissions	Patrik	Matej Halinkovič	Done	Done	03/03/2022 11:08	10/03/2022 09:37
Task	PP-152	Implement CI/CD for webapp	Tibor Sloboda	Matej Halinkovič	Done	Done	03/03/2022 10:47	06/03/2022 10:22
Task	PP-151	Update /phototox to new version	Tibor Sloboda	Matej Halinkovič	Done	Done	03/03/2022 10:47	06/03/2022 10:22
Task	PP-150	Create storage for submitted data	Matej Halinkovič	Matej Halinkovič	Done	Done	03/03/2022 10:38	10/03/2022 09:40
Task	PP-149	Django user account creation	Kateřina Mušková	Matej Halinkovič	Done	Done	03/03/2022 10:17	10/03/2022 09:45
Task	PP-148	Convert molecules to MOL	Tibor Sloboda	Matej Halinkovič	Done	Done	03/03/2022 10:12	10/03/2022 09:12
Task	PP-147	Sprint retrospective 7	Jakub Knánik	Matej Halinkovič	Done	Done	03/03/2022 10:10	10/03/2022 09:26
Task	PP-146	Meeting report 3	Matej Halinkovič	Matej Halinkovič	Done	Done	03/03/2022 10:10	05/03/2022 19:33
Task	PP-144	Create academic domain table	Kateřina Mušková	Matej Halinkovič	Done	Done	26/02/2022 19:02	06/03/2022 17:23
Task	PP-136	Provide PT/NPT source	Matej Halinkovič	Matej Halinkovič	Done	Done	24/02/2022 11:35	05/03/2022 19:33

Šprint č.9

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-139	Extract data for current molecules from Alvadesc	Tibor Sloboda	Matej Halinkovič	Done	Done	24/02/2022 11:47	11/03/2022 21:14
Task	PP-140	Alvadesc training experiments	Matej Halinkovič	Matej Halinkovič	Done	Done	24/02/2022 11:47	16/03/2022 12:47
Task	PP-156	Integrate Login/Register to existing webapp	Jakub Knánik	Matej Halinkovič	Done	Done	03/03/2022 11:05	16/03/2022 12:44
Task	PP-158	Enable SSL for webapp and website	Tibor Sloboda	Tibor Sloboda	Done	Done	05/03/2022 05:33	11/03/2022 21:14
Task	PP-160	Set-up django tests	Kateřina Mušková	Tibor Sloboda	Done	Done	06/03/2022 14:45	16/03/2022 12:46
Task	PP-161	Set-up django automated testing and integration via G-Hub	Tibor Sloboda	Tibor Sloboda	Done	Done	06/03/2022 14:46	16/03/2022 13:36
Task	PP-162	Meeting report 4	Kateřina Mušková	Matej Halinkovič	Done	Done	10/03/2022 10:19	16/03/2022 12:46
Task	PP-163	Sprint retro 8	Jakub Maruniak	Matej Halinkovič	Done	Done	10/03/2022 10:19	19/03/2022 11:35
Task	PP-164	Write Django authentication backend	Patrik	Matej Halinkovič	Done	Done	10/03/2022 10:29	16/03/2022 12:47

Šprint č.10

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-153	Data augmentation script	Tibor Sloboda	Matej Halinkovič	Done	Done	03/03/2022 10:53	24/03/2022 10:15
Task	PP-154	Create page for submitted data validation	Patrik	Matej Halinkovič	Done	Done	03/03/2022 10:53	24/03/2022 10:21
Task	PP-166	Django unit tests	Kateřina Mušková	Matej Halinkovič	Done	Done	16/03/2022 13:37	24/03/2022 10:16
Task	PP-167	Sprint retro 9	Jakub Knánik	Matej Halinkovič	Done	Done	16/03/2022 13:38	24/03/2022 10:15
Task	PP-168	Meeting report 5	Kateřina Mušková	Matej Halinkovič	Done	Done	16/03/2022 13:38	24/03/2022 10:29
Task	PP-169	IIT.SRC paper draft	Matej Halinkovič	Matej Halinkovič	Done	Done	19/03/2022 11:36	24/03/2022 10:22

Šprint č.11

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-170	Finalize IIT.SRC paper	Matej Halinkovič	Matej Halinkovič	Done	Done	24/03/2022 11:34	29/03/2022 23:41
Task	PP-171	Meeting report 6	Matej Halinkovič	Matej Halinkovič	Done	Done	24/03/2022 11:34	31/03/2022 10:26
Task	PP-172	Sprint retro 10	Jakub Knánik	Matej Halinkovič	Done	Done	24/03/2022 11:34	31/03/2022 10:12
Task	PP-173	Finish test suite	Tibor Sloboda	Matej Halinkovič	Done	Done	24/03/2022 11:36	29/03/2022 23:42
Task	PP-174	English/Slovak localization	Jakub Knánik	Matej Halinkovič	Done	Done	24/03/2022 11:37	31/03/2022 10:12
Task	PP-175	Implement submitted data validation backend	Patrik	Matej Halinkovič	Done	Done	24/03/2022 11:37	31/03/2022 10:26
Task	PP-176	Implement TaBnet	Tibor Sloboda	Matej Halinkovič	Done	Done	24/03/2022 11:44	29/03/2022 23:42
Task	PP-177	Start user input validation	Jakub Knánik	Matej Halinkovič	Done	Done	24/03/2022 11:46	31/03/2022 10:12

Šprint č.12

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-179	Create simple module for fast experiments of our current best model	Matej Halinkovič	Tibor Sloboda	Done	Done	29/03/2022 15:33	01/04/2022 21:15
Task	PP-180	Upload this semester's documents to team website	Patrik	Matej Halinkovič	Done	Done	29/03/2022 23:16	07/04/2022 10:28
Task	PP-181	User input validation	Jakub Knánik	Matej Halinkovič	Done	Done	31/03/2022 10:13	06/04/2022 12:51
Task	PP-182	Update our website's model to best version	Matej Halinkovič	Matej Halinkovič	Done	Done	31/03/2022 10:22	06/04/2022 12:50
Task	PP-184	Meeting report 7	Matej Halinkovič	Matej Halinkovič	Done	Done	31/03/2022 10:23	31/03/2022 12:50
Task	PP-185	Sprint retro 11	Jakub Knánik	Matej Halinkovič	Done	Done	31/03/2022 10:23	06/04/2022 12:51
Task	PP-186	Create backend separation for validated data	Patrik	Matej Halinkovič	Done	Done	31/03/2022 10:29	02/04/2022 12:02
Task	PP-187	Implement chemist accounts	Patrik	Matej Halinkovič	Done	Done	31/03/2022 10:30	02/04/2022 12:02
Task	PP-188	Fix tests in issues	Kateřina Mušková	Matej Halinkovič	Done	Done	31/03/2022 10:37	07/04/2022 10:28
Task	PP-189	Find the best option for optimising experiments	Tibor Sloboda	Matej Halinkovič	Done	Done	31/03/2022 10:39	01/04/2022 18:23
Task	PP-190	Implement the best option for optimising experiments	Tibor Sloboda	Matej Halinkovič	Done	Done	31/03/2022 10:39	02/04/2022 12:02
Task	PP-192	Prepare MLOps infrastructure	Kateřina Mušková	Matej Halinkovič	Done	Done	31/03/2022 10:49	07/04/2022 10:28

Šprint č.13

Issue Type	Key	Summary	Assignee	Reporter	Status	Resolution	Created	Resolved
Task	PP-203	Spiritual journey to find better web app design	Kateřina Mušková	Matej Halinkovič	Done	Done	07/04/2022 11:00	14/04/2022 10:42
Task	PP-202	Add limit to prediction	Patrik	Matej Halinkovič	Done	Done	07/04/2022 10:50	14/04/2022 10:43
Task	PP-201	MLops model training and comparison	Kateřina Mušková	Matej Halinkovič	Done	Done	07/04/2022 10:29	14/04/2022 10:50
Task	PP-200	Calculation in progress indicator	Jakub Knánik	Matej Halinkovič	Done	Done	07/04/2022 10:26	14/04/2022 10:43
Task	PP-199	Initial video design	Tibor Sloboda	Matej Halinkovič	Done	Done	07/04/2022 10:13	14/04/2022 10:41
Task	PP-197	Sprint retro 12	Jakub Knánik	Matej Halinkovič	Done	Done	07/04/2022 10:12	14/04/2022 10:43
Task	PP-196	Meeting report 8	Kateřina Mušková	Matej Halinkovič	Done	Done	07/04/2022 10:12	14/04/2022 10:43
Task	PP-195	Initial poster design	Matej Halinkovič	Matej Halinkovič	Done	Done	07/04/2022 10:11	14/04/2022 10:41
Task	PP-194	Fix incompatible features with new deployed model	Matej Halinkovič	Tibor Sloboda	Done	Done	07/04/2022 09:35	14/04/2022 10:42
Task	PP-191	Expanded GCN experiments	Matej Halinkovič	Matej Halinkovič	Done	Done	31/03/2022 10:42	14/04/2022 10:42

PRÍLOHA J EXPORT ÚLOH

J-16

Šprint č.14

Issue Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
Task	PP-219	Switch Random Forest to XGBoost for Rdkit Tune experiments	Tibor Sloboda	Tibor Sloboda	Medium	Done	Done	14/04/2022 10:59:43	15/04/2022 22:22:03
Task	PP-218	Implement MLops tests	Kateřina Mušková	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:54:24	18/04/2022 12:56:44
Task	PP-217	Integrate MLops process with database	Kateřina Mušková	Tibor Sloboda	Medium	Done	Done	14/04/2022 10:54:05	20/04/2022 12:08:59
Task	PP-216	Display which SMILES code was incorrect	Patrik	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:53:26	21/04/2022 10:33:43
Task	PP-215	Continued GNN experiments	Matej Halinkovič	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:51:34	20/04/2022 11:59:06
Task	PP-214	Finalize IIT.SRC poster	Matej Halinkovič	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:51:18	18/04/2022 16:42:34
Task	PP-213	Meeting report 9	Tibor Sloboda	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:51:14	19/04/2022 10:17:19
Task	PP-212	Sprint retro 13	Tibor Sloboda	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:51:09	19/04/2022 10:17:18
Task	PP-211	Domain request form	Patrik	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:50:05	21/04/2022 10:33:40
Task	PP-210	Add submitted data	Patrik	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:49:43	21/04/2022 10:23:09
Task	PP-208	URL field explanation	Jakub Knánik	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:47:41	21/04/2022 10:20:54
Task	PP-207	User page	Jakub Knánik	Matej Halinkovič	Medium	Done	Done	14/04/2022 10:46:42	21/04/2022 10:20:49

Šprint č.15

Issue Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
Task	PP-231	Add delete flag to suggestions	Jakub Knánik	Tibor Sloboda	Medium	Done	Done	20/04/2022 13:26	27/04/2022 11:54
Task	PP-230	ESTIV Abstract	Matej Halinkovič	Matej Halinkovič	Medium	Done	Done	20/04/2022 13:23	27/04/2022 11:54
Task	PP-229	Populate_light - storage of displayed descriptors	Kateřina Mušková	Tibor Sloboda	Medium	Done	Done	20/04/2022 13:23	27/04/2022 11:54
Task	PP-228	Touchl T Blogpost	Jakub Knánik	Matej Halinkovič	Medium	Done	Done	20/04/2022 13:00	27/04/2022 12:05
Task	PP-227	Create new banner for mltox	Tibor Sloboda	Matej Halinkovič	Medium	Done	Done	20/04/2022 12:33	22/04/2022 12:24
Task	PP-226	GNN attention pooling	Tibor Sloboda	Matej Halinkovič	Medium	Done	Done	20/04/2022 12:22	27/04/2022 11:54
Task	PP-225	Optuna single objective	Tibor Sloboda	Matej Halinkovič	Medium	Done	Done	20/04/2022 12:22	27/04/2022 11:54
Task	PP-224	Video post-production	Tibor Sloboda	Matej Halinkovič	Medium	Done	Done	20/04/2022 12:20	27/04/2022 11:54
Task	PP-223	Sprint retro 14	Kateřina Mušková	Matej Halinkovič	Medium	Done	Done	20/04/2022 12:19	27/04/2022 11:54
Task	PP-222	Meeting report 10	Kateřina Mušková	Matej Halinkovič	Medium	Done	Done	20/04/2022 12:19	27/04/2022 11:54

K Prihláška na TP Cup

Kto sme?

Náš tím je tvorený ľuďmi, ktorí majú blízky vzťah k programovaniu, strojovému učeniu a dátovej analýze. Zo školských ale aj pracovných projektov sme získali nielen technické, ale aj manažérské skúsenosti. Každý člen tímu sa stretol s výskumnou prácou – vieme efektívne vyhľadávať zdroje a získavať z nich relevantné informácie, dokážeme výskumný projekt dotiahnuť do prezentovateľnej podoby a niektorí z nás už výskumnú prácu aj publikovali. Ak sa stretneme so zdanlivo neriešiteľným problémom, sme silne motivovaní ho dôkladne analyzovať a nájsť vhodné riešenie. Práve preto si myslíme, že sme tí praví pre riešenie tohto projektu.

Kontext projektu a motivácia

V posledných rokoch môžeme pozorovať zvýšenú prevalenciu kožných ochorení. Jednou z príčin je aj časté používanie rôznych liekov či kozmetiky a celková chemizácia života (potravinové aditíva, rastlinné zložky, umelé sladidlá, ...). Zvýšená citlivosť kože je priamo spôsobená interakciou žiarenia a chemickej látky v koži, ktorá je schopná pohlcovať žiarenie. Z toho dôvodu je nutné testovať všetky chemické substancie prichádzajúce na trh na fototoxicitu (FOTOTOX). V minulosti sa FOTOTOX testovala výlučne na zvieracích modeloch, čo je nehumánne a zároveň nepresné vzhľadom na rozdielnosť kože ľudí a zvierat. V súčasnosti sa FOTOTOX testuje na bunkových kultúrach (OECD guideline 432) a na tkanivových kultúrach (OECD 498). Pre stanovenie dermatotoxicity je možné podľa guidelinov OECD využívať aj predikcie pomocou informačných technológií, ktoré skracujú čas, a šetria obrovské finančné prostriedky na "mokrý" experiment. Avšak pre stanovenie FOTOTOX takáto možnosť chýba. Preto je hlavným cieľom predkladaného projektu navrhnuté riešenie pre in silico experimenty na stanovenie FOTOTOX. Myslíme si, že náš projekt by mohol pomôcť pri tvorení nových štandardov OECD. Výsledky projektu navyše budú pomáhať chrániť zdravie ako aj znížovať náklady pri vyhodnocovaní fototoxicity látok, preto túto tému považujeme za o to viac dôležitú, a zaujímavú.

Testovanie na zvieratách je žiaľ aj dnes súčasťou vývoja rôznych produktov - liekov, kozmetiky či farbív. Hoci spočiatku je možné testovať len

na odobratom tkanive, celkový efekt vyvájaných produktov ukážu až testy na žijúcich zvieratách. Takéto testy ukážu, ako zareaguje koža po aplikovaní kozmetiky alebo akým spôsobom lieky ovplyvňujú orgány v tele. Ak testovaný produkt prejde bezpečnostnými testami a ukáže sa, že zabral na testovaných zvieratách, môže byť ďalej študovaný na ľuďoch. Tento proces je však už na dnešné pomery zastaraný a tak ako aj v iných odvetviach, tak aj v tejto oblasti je načase sa pohnúť vpred. Testovanie na zvieratách je totiž nielen nehumánne, ale aj neekologicke a finančne nákladné. Každým rokom sa na testovanie použijú milióny zvierat rôznych druhov (myši, zajace, opice, psi, ...), pričom takéto testy sa často vykonávajú len preto, že to požadujú kontrolné úrady či legislatíva niektorých krajín.

Dnes už pritom existujú plnohodnotné alternatívy voči testovaniu na zvieratách. Napríklad s bunkovými kultúrami vieme vytvoriť miniatúry ľudských orgánov, na ktorých možno simulovať vplyv liekov. V prípade fototoxicity sa môžu použiť bunky z radu 3T3, ktoré sú vystavené testovanej chemikálii bez/s prítomnosťou svetla. Ďalšou alternatívou k testovaniu na zvieratách sú aj počítačové modely. Za posledné obdobie vývoj v tejto oblasti vzrástol enormným tempom, aj vďaka skúmaniu a využitiu umelej inteligencie. Jednou z možností využitia technológií je dolovanie v dátach, pomocou ktorého vieme vytvoriť predpovede o možnom nebezpečenstve chemických látok na základe už existujúcich dát. Veríme tak, že projekty ako je tento, pomôžu v procese prechodu od testovania na zvieratách k alternatívnym metódam.

Cieľ projektu

V tomto projekte sa zameriame na jednu oblasť nežiadúcich vplyvov liekov a látok, konkrétnie na fototoxicitu. Fototoxicická je látka, ktorá je toxicá iba po pôsobení svetla. Cieľom projektu je využitie modelov kvantitatívnej závislosti aktivity na štruktúre (QSAR) pre získanie vzťahov medzi chemickou štruktúrou a biologickou aktivitou v súbore dát a následné predpovedanie FOTOtox aktivity nových chemických látok. Na základe zhromaždených dát a natrénovanému modelu tak bude vedieť naše riešenie predikovať fototoxicitu látky. Nakoniec celé riešenie nasadíme takým spôsobom, aby používateľ (toxikológ, chemik) mohol jednoducho podľa zápisu chemickej látky predikovať informácie o jej fototoxicite.

Návrh riešenia

Aby sme mohli model úspešne vytvoriť, potrebujeme najskôr zhromaždiť dostatočnú vzorku dát, z ktorej budeme môcť určiť, či niektorá konkrétna látka je alebo naopak nie je fototoxiccká. Analyzujeme tak veľké množstvo publikácií a relevantné informácie extrahujeme. Na identifikáciu chemickej látky vieme použiť CAS identifikačné kódy. Automatickým procesom dokážeme získať informácie o chemických látkach z verejne prístupnej databázy chemikálií, akou je napríklad PubChem.

Z týchto informácií nám vyplynie množstvo molekulových deskriptorov - potencionálnych prediktorov. Získané dáta následne analyzujeme technikou EDA, podľa ktorej budeme môcť identifikovať možné vzory v dátach a testovať hypotézy. Na základe hodnôt korelácií premenných môžeme pozorovať, ktoré vlastnosti látok spolu súvisia a akým spôsobom vplyvajú na predikovanú premennú, teda na fototoxicitu. Medzi získanými dátami o chemických látkach sú aj ich SMILES kódy. Ide o chemický zápis, ktorý umožňuje reprezentáciu chemickej štruktúry spôsobom, aby ju mohol využiť počítač. Tieto kódy vieme takisto analyzovať a hľadať v nich vzory, ktoré by mohli vplyvať na fototoxicitu, napr. počet aromatických cyklov. Ďalej sa zameriame na strojové učenie, preskúmame fungovanie rôznych algoritmov, ktoré by sa pre nás účel mohli hodíť (napr. v projekte pracujeme s chemickou štruktúrou, jednou z techník, ktoré by sme mohli aplikovať, je model grafovej neurónovej siete). Postupným procesom vytvoríme čo najúspešnejší a najpresnejší model, ktorý potom použijeme vo výslednom riešení. Finálna aplikácia môže mať podobu webovej aplikácie, do ktorej používateľ zadá SMILES kód chemickej látky a program vráti informácie o jej fototoxicite. Uvedené informácie môže následne používateľ ďalej využiť pre potreby svojej vlastnej práce. Týmto spôsobom tak pomáhame nielen v tvorbe alternatívy k testovaniu na zvieratách, ale aj v ďalšom výskume a pri tvorbe nových postupov OECD.

L Motivačný dokument

Tím

Náš tím je skupina vysoko motivovaných ľudí s blízkym vzťahom k programovaniu, strojovému učeniu a dátovej analytike. Pozostáva z bakalárov Mateja, Tibora, Jakuba K., Jakuba M., Patrika a Katky, ktorí sa navzájom dopĺňajú v znalostiach z rôznych oblastí informatiky. Po technickej stránke nájdeme v našej skupine znalosti zo všetkých potrebných oblastí. Skúsenosti z manažovania má každý člen tímu, nadobudnuté či už zo školských alebo aj pracovných projektov vo väčších tímov. V rámci rôznych tímov, ktoré boli vedené agilným spôsobom, sme sa všetci stretli s verzovacími nástrojmi (Git), task manažérmi (Jira, Trello, Shortcut) alebo CI/CD nástrojmi (Jenkins).

Jakub M. má dva roky skúseností v agile tíme na pozícii Application Developer, kde na projekte v IBM pracoval na testovacích pipelinách (automatic functional, performance, API) pre webovú aplikáciu. Matej pracuje v oblasti Data Science / ML engineering pre spoločnosť Dell Technologies - pracovná náplň zahrňa spracovanie big data, budovanie efektívnych modelov strojového učenia a ich uvedenie do produkcie. Katka má skúsenosť z výskumného tímu VeriFIT na FIT VUT, ktorý se zaoberá verifikáciou a testováním systémov.

Rovnako tak väčšina z nás ovláda základy databázových systémov. Na vyššej úrovni potom ovláda Matej, ktorý má zároveň pracovné skúsenosti s analytickými OLAP databázovými technológiami Greenplum, Hadoop a Hive. S dátami je neodmysliteľne spojená aj manipulácia s nimi a ich analýza, s čím máme všetci v tíme aspoň základné skúsenosti. Väčšia časť tímu z bakalárskeho štúdia (predmet Inteligentná Analýza Údajov), a Katka z predmetu Zpracování a vizualizace dat v prostredí Python na VUT FIT.

Celý tím má skúsenosti s výskumnou prácou (prevažne z bakalárskych prác), kde viaceri členov tímu pracovalo na témach z oblasti umelej inteligencie a hlbokého učenia. Traja členovia tímu boli súčasťou výskumnej orientácie, v rámci ktorej svoje výstupy verejne publikovali. Jakub K. sa venoval Analýze problematiky nepravdivých informácií. Matej sa so svojou bakalárskou prácou Analýza hudby pomocou umelých neurónových sietí zúčastnil konferencie IIT.SRC. Tibor má publikáciu v žurnáli Diagnostics vo spolupráci s doktorandom fakulty a zahraničnými lekármi v oblasti diagnos-

tiky pomocou hlbokého učenia.

Každý člen tímu má skúsenosti s technickým dokumentovaním kódu a so softvérovým inžinierstvom a architektúrou. Preferovaným jazykom tímu je Python, kde máme všetci aspoň rudimentárne schopnosti aj z testovania a dokumentácie. Dodatočne má Tibor skúsenosti s kontrolou kvality kódu, a významnú znalosť noriem PEP kvôli kontribúciam do Python codebase. Katka má taktiež rozsiahlejšie znalosti z oblasti testovania a dynamickej analýzy, okrem iného s unit-testami v Pythone a C++, ale aj s integračnými testami v Bashi. Všetci členovia tímu majú zapísané predmety na inžinierskom štúdiu z oblasti umelej inteligencie, akými sú Objavovanie znalostí (všetci), Neurónové siete (všetci), Digitálne spracovanie zvuku, obrazu a biosignálov (Matej, Jakub K., Jakub M.) a Vyhľadávanie informácií (Matej, Patrik, Tibor).

Motivácia

11. (Q)SAR analýza fototoxickej látok

Ochrana životného prostredia je v posledné roky významnou a dôležitou tému. Čoraz viac firiem sa snažia urobiť svoje business procesy ekologickejšie a to ako z dôvodu ekonomickeho tak aj sociálneho. Tento projekt vidíme ako šancu pomôcť tejto iniciatíve a pracovať na téme, ktorú všetci považujeme za nesmierne dôležitú.

Rôzne farmaceutické, kozmetické a potravinárske firmy dodnes používajú testy na zvieratách alebo rôzne drahé experimenty na tkanivách biopsovaných zo zvierat aby overili fototoxicitu látok. Je to nielen ne-ekologické, nehumánne ale aj finančne náročné, a chceme preto revolucionizovať spôsob determinácie fototoxicity v látkach.

Nakoľko je projekt riešený v spolupráci s organizáciou OECD, jeho praktické použitie v nadväzujúcich projektoch v rozvinutých ale aj rozvojových krajinách môže mať z dlhodobého hľadiska citelný dopad na ekológiu a zelenú budúcnosť. Nás tím je zručný v oblasti spracovania signálu, analýzy dát a strojového učenia, a má skúsenosti vo výskume v oblasti bioinformatiky. Dodatočne, v našom tíme máme členov so zapísaným predmetom Digitálne spracovanie zvuku, obrazu a biosignálov, ktorý je obsahom priamo relevantný k zadanej téme.

15. ION Mobility Spectrometry for Rapid HEMP potency testing

Vo využití CBD pre medicínske účely vidíme veľký potenciál a v rámci tímu s ním máme aj osobné skúsenosti. Radi by sme pomohli rozvoju tejto potencionálne život-meniacej oblasti a zároveň prehĺbili naše vedomosti nielen v rámci súvisiacich technológií, ale aj v problémoch, s ktorými sa musia pestovatelia vysporiadátať.

V dnešnej dobe sa v poľnohospodárstve už bežne využívajú podobné prístroje meracie napríklad obsah cukru v hrozne, obsah lepku v obilí, alebo kvalita medu. Týmto projektom by sme teda radi rozšírili už existujúce riešenia, ktoré sú v dnešnej dobe nepostrádateľné.

Matej, Tibor, a Jakub majú skúsenosti so spracovaním obrazu a signálov z bakalárskych prác. Počas inžinierskeho štúdia absolvujeme všetky predmety spojené s danou térou, niektorí členovia tímu majú tiež skúsenosti aj s budovaním experimentov pre porovnávanie vhodnosti rôznych prístupov strojového učenia a ich použitia v praxi.

Spracovanie signálu má bohaté využitie, preto by sme ocenili príležitosť nadobudnúť počas Tímového projektu relevantné vedomosti, ktoré by nám v budúcnosti mohli otvoriť dvere k množstvu ďalších príležitostí.

4. Adverse Media Screening

V modernej dobe IOT zariadení a prepojenosti je takmer nemožné vyhnúť sa zanechaniu rôznych digitálnych stôp na webe. Minulosť jednotlivcov sa čoraz jednoduchšie preveruje a skrytie spojenia s nelegálnymi aktivitami je čoraz náročnejšie.

Takáto previerka je však pre spoločnosti veľmi nákladná, nakolko z veľkej časti pozostáva z manuálneho prehľadávania online zdrojov a digitálnych záznamov za účelom vytvárania správ o histórii osoby alebo spoločnosti. Tieto správy sú však nevyhnutné pre identifikovanie potencionálnych ukazovateľov podvodov, prania peňazí a iných nepriaznivých aktivít.

Našim cieľom je využiť nadobudnuté skúsenosti v oblasti strojového učenia pre automatizáciu procesu prehľadávania zdrojov a preveriť zapojenie osôb a spoločností do nelegálnych aktivít. Nami vytvorené riešenie by mohlo výrazne znížiť množstvo času a peňažných prostriedkov potrebných na vykonávanie tejto dôležitej činnosti.

Člen tímu Jakub K. má skúsenosti s technológiami relevantnými pre túto tému, a to konkrétnie so spracovaním prirodzeného jazyka, web scraping a web

crawling. Zvyšok tímu je skúsený v oblasti analýzy dát a strojového učenia. Všetci si chceme prehľbiť naše vedomosti v oblasti spracovania prirodzeného jazyka, KYC noriem a webových technológií.

Poradie tém

1 - najpreferovanejšie

1. 11. (Q)SAR analýza fototoxických látok
2. 15. Ion Mobility Spectrometry for Rapid HEMP Potency Testing
3. 4. Adverse Media Screening
4. 3. DataHub pre rôzne typy zariadení, ich spracovanie / analýzu / vizualizáciu
5. 12. Spektrometrické rozpoznávanie túh do pera
6. 19. Automatizácia procesov KYC (Know your client) a AML (Anti-money laundering)
7. 14. IoT platforma na priemyselnú automatizáciu - malý pivovar
8. 5. Vytvorenie inteligentného model-based agenta (umelá inteligencia na báze Knowledge grafov) pre tvorbu komplexných dátových štruktúr a vzťahov pre aplikovaný výskum v klinickej onkológii
9. 8. Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU]
10. 13. Navigácia v smartfóne pomocou rozšírenej reality
11. 9. Monitorovanie a správa systému pre výrobný areál [LOMON]
12. 7. Vizualizácia softvéru vo virtuálnej a rozšírenej realite [VizReal]

M Protokol z testovania

1. Fototox (tím č. 20): Posudok na prototyp tímu č. 20: Fototox

Tím 20 sa zaoberl predikciou fototoxických látok metódou QSAR. V poslednej dobe sa kladie zvýšený dôraz na identifikáciu potenciálnych fototoxických látok, ktoré môžu mať negatívny vplyv nielen na ľudské zdravie ale aj na zdravie živočíchov. Avšak testovanie na úrovni in vitro je veľmi zdíľavé a finančne náročné. Práve in silico metódy a metódy QSAR sú preto ideálou alternatívou k in vitro testom. Tento spôsob predstavuje rýchlu a lacnú alternatívu. Dopolň však nie je dostupná aplikácia umožňujúca spoľahlivú predikciu. Výsledkom práce tímu 20 je webová aplikácia umožňujúca online predikciu fototoxicity na základe SMILES kódov. Prevedenie aplikácie je intuitívne a umožňuje rýchle získanie výsledkov. Web aplikácia umožňuje nielen predikovať fototoxicitu ale aj spolupracovať na vylepšenie modelu spolu s tvorcami. Odborným užívateľom je umožnené sa registrovať na stránke a pridávať nové experimentálne dáta. Táto možnosť je však umožnená len registrovaným užívateľom, ktorí sú odborníkmi v danej oblasti. Aplikácia poskytuje aj výpočet základných vlastností molekúl na základe SMILES kódov.

V literatúre sa uvádzaj iba jedna QSAR analýza na predikciu fototoxicity látok. Ich úspešnosť je 85 % senzitivita a 70% špecificka. Oproti tomu aplikácia Tímu 20 ponúka úspešnosť prediekcie na úrovni 88% senzitivita a 75% špecificka. Preto predložený projekt význam môže napomôcť výskumu a preidkcií fototoxicity.

Návrh na zlepšenie:

1. Pre možnosť väčšej flexibility vo forme zadávania štruktúr by som odporučila implementovať aj nástroj na rýchle kreslenie chemických štruktúr a CAS.
2. Napojenie sa na voľne prístupné databázy.
3. Pripraviť aplikáciu aj pre neodborných používateľov. Napríklad výsledok predikcie je PT resp. NPT. Neodborník v danej oblasti by mohol mať problém s pochopením výsledku predikcie.



Dr.rer.nat., Ing. Helena Kandárová, ERT
Prezident Európskej spoločnosti pre in vitro toxikológiu

N Generovaná technická dokumentácia

DataPipeline

Generated by Doxygen 1.8.17

1 Usage:	1
2 Namespace Index	3
2.1 Packages	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 run_pipeline Namespace Reference	11
6.1.1 Detailed Description	11
6.1.2 Function Documentation	11
6.1.2.1 run_pipe()	11
6.2 scripts Namespace Reference	11
6.3 scripts.cleaner Namespace Reference	12
6.3.1 Detailed Description	12
6.3.2 Function Documentation	12
6.3.2.1 check_correlated_column()	13
6.3.2.2 check_outliers()	13
6.3.2.3 create_correlation_graphs()	13
6.3.2.4 get_correlated_descriptors()	14
6.3.2.5 get_duplicit_correlated_descriptors()	14
6.3.2.6 load_data()	14
6.3.2.7 main()	14
6.3.2.8 plot_correlation_heatmap()	15
6.3.2.9 process_config()	15
6.3.2.10 remove_duplicits()	15
6.3.2.11 remove_useless_columns()	15
6.3.2.12 save_mask()	16
6.3.2.13 set_proper_data_type()	16
6.3.2.14 unique_count()	16
6.3.3 Variable Documentation	16
6.3.3.1 args	16
6.3.3.2 input_data	17
6.3.3.3 output_data	17
6.3.3.4 project_dir	17
6.3.3.5 script_dir	17
6.4 scripts.crawlers Namespace Reference	17

6.5 scripts.crawlers.abstract_crawler Namespace Reference	17
6.5.1 Detailed Description	17
6.6 scripts.crawlers.crawler_orchestrator Namespace Reference	18
6.6.1 Detailed Description	18
6.6.2 Function Documentation	18
6.6.2.1 main()	18
6.7 scripts.crawlers.dependencies Namespace Reference	18
6.8 scripts.crawlers.dependencies.pubchem Namespace Reference	18
6.8.1 Detailed Description	18
6.9 scripts.crawlers.dependencies.swissadme Namespace Reference	19
6.9.1 Detailed Description	19
6.10 scripts.crawlers.dependencies.swisstarget Namespace Reference	19
6.10.1 Detailed Description	19
6.11 scripts.merger Namespace Reference	19
6.11.1 Detailed Description	19
6.11.2 Function Documentation	19
6.11.2.1 merge()	19
6.12 scripts.populate_chem Namespace Reference	20
6.12.1 Detailed Description	20
6.12.2 Function Documentation	20
6.12.2.1 populate()	20
7 Class Documentation	21
7.1 scripts.crawlers.abstract_crawler.AbstractCrawler Class Reference	21
7.1.1 Detailed Description	22
7.1.2 Member Function Documentation	22
7.1.2.1 crawl()	22
7.1.2.2 is_file_downloaded()	22
7.2 scripts.crawlers.dependencies.pubchem.PubchemCrawler Class Reference	23
7.2.1 Detailed Description	24
7.2.2 Member Function Documentation	24
7.2.2.1 crawl()	24
7.2.2.2 get_classname()	24
7.2.2.3 is_file_downloaded()	24
7.2.3 Member Data Documentation	25
7.2.3.1 COLUMNS	25
7.3 scripts.crawlers.dependencies.swissadme.SwissadmeCrawler Class Reference	25
7.3.1 Detailed Description	26
7.3.2 Member Function Documentation	26
7.3.2.1 crawl()	26
7.3.2.2 get_classname()	27
7.3.2.3 is_file_downloaded()	27

7.4 scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler Class Reference	27
7.4.1 Detailed Description	28
7.4.2 Member Function Documentation	28
7.4.2.1 crawl()	28
7.4.2.2 get_classname()	29
7.4.2.3 is_file_downloaded()	29
7.4.3 Member Data Documentation	29
7.4.3.1 COLUMNS	29
8 File Documentation	31
8.1 README.md File Reference	31
8.2 run_pipeline.py File Reference	31
8.3 scripts/__init__.py File Reference	31
8.4 scripts/crawlers/__init__.py File Reference	31
8.5 scripts/crawlers/dependencies/__init__.py File Reference	31
8.6 scripts/cleaner.py File Reference	32
8.7 scripts/crawlers/abstract_crawler.py File Reference	32
8.8 scripts/crawlers/crawler_orchestrator.py File Reference	33
8.9 scripts/crawlers/dependencies/pubchem.py File Reference	33
8.10 scripts/crawlers/dependencies/swissadme.py File Reference	33
8.11 scripts/crawlers/dependencies/swisstarget.py File Reference	33
8.12 scripts/merger.py File Reference	34
8.13 scripts/populate_chem.py File Reference	34
Index	35

Chapter 1

Usage:

Run the file [run_pipeline.py](#) after installing python using conda from environment.yml

[run_pipeline.py](#) supports the following arguments:

- **-inputfile "path"** to specify .csv containing [phototoxic, name] pairs, where phototoxic is either 0 or 1
- **-outputfile "path"** to specify final output for .csv file
- **-noscrape** to skip the scraping using crawlers

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

run_pipeline	11
scripts	11
scripts.cleaner	12
scripts.crawlers	17
scripts.crawlers.abstract_crawler	17
scripts.crawlers.crawler_orchestrator	18
scripts.crawlers.dependencies	18
scripts.crawlers.dependencies.pubchem	18
scripts.crawlers.dependencies.swissadme	19
scripts.crawlers.dependencies.swisstarget	19
scripts.merger	19
scripts.populate_chem	20

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ABC	
scripts.crawlers.abstract_crawler.AbstractCrawler	21
scripts.crawlers.dependencies.pubchem.PubchemCrawler	23
scripts.crawlers.dependencies.swissadme.SwissadmeCrawler	25
scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler	27

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

scripts.crawlers.abstract_crawler.AbstractCrawler	21
scripts.crawlers.dependencies.pubchem.PubchemCrawler	23
scripts.crawlers.dependencies.swissadme.SwissadmeCrawler	25
scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler	27

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

run_pipeline.py	31
scripts/__init__.py	31
scripts/cleaner.py	32
scripts/merger.py	34
scripts/populate_chem.py	34
scripts/crawlers/__init__.py	31
scripts/crawlers/abstract_crawler.py	32
scripts/crawlers/crawler_orchestrator.py	33
scripts/crawlers/dependencies/__init__.py	31
scripts/crawlers/dependencies/pubchem.py	33
scripts/crawlers/dependencies/swissadme.py	33
scripts/crawlers/dependencies/swisstarget.py	33

Chapter 6

Namespace Documentation

6.1 run_pipeline Namespace Reference

Functions

- def [run_pipe\(\)](#)

6.1.1 Detailed Description

Manages the execution of the pipeline.

6.1.2 Function Documentation

6.1.2.1 [run_pipe\(\)](#)

```
def run_pipeline.run_pipe( )
```

Runs the pipeline in sequence of steps: Crawler, merger, chem populator, cleaner.

The chem populator also merges existing data with new data before it's sent to the cleaner.

Multiple arguments are supported, as described in README.

6.2 scripts Namespace Reference

Namespaces

- [cleaner](#)
- [crawlers](#)
- [merger](#)
- [populate_chem](#)

6.3 scripts.cleaner Namespace Reference

Functions

- pd.DataFrame `load_data` (str csv_file_name)
- pd.DataFrame `set_proper_data_type` (pd.DataFrame df)
- [str, int] `unique_count` (pd.Series column)
- pd.DataFrame `remove_useless_columns` (pd.DataFrame df)
- list `get_duplicat_correlated_descriptors` (nx.MultiGraph graphs, exclude)
- def `plot_correlation_heatmap` (pd.DataFrame correlations, str fig_location, int threshold)
- def `get_correlated_descriptors` (pd.DataFrame df, int threshold, str fig_location)
- list `create_correlation_graphs` (pd.DataFrame correlations, str fig_location, float threshold)
- None `save_mask` (pd.DataFrame df, str mask_name, [None, list] columns=None, [None, list] rows=None, str data_dir="data")
- pd.DataFrame `check_correlated_column` (pd.DataFrame df, float threshold=0.9, bool remove=False, list preserve_columns=None, str plot_dir="plot", str data_dir="data")
- pd.DataFrame `remove_duplicits` (pd.DataFrame df, str subset, keep="first")
- def `check_outliers` (pd.DataFrame df, float threshold=4.2, bool remove=False)
- None `main` (str input_file="data/chem_output/chem_populated.csv", str output_file="data/final/phototox.csv", float correlation_threshold=0.95, float outliers_threshold=4.2, list preserve_columns=None, bool remove=False)
- dict `process_config` (str config_file="conf/cleaner.ini")

Variables

- `script_dir` = os.path.dirname(os.path.realpath(__file__))
- `project_dir` = os.path.dirname(`script_dir`)
- dict `args` = `process_config`(os.path.join(`project_dir`, "conf/cleaner.ini"))
- `input_data` = os.path.join(`project_dir`, "data/chem_output/chem_populated.csv")
- `output_data` = os.path.join(`project_dir`, "data/chem_output/phototox.csv")

6.3.1 Detailed Description

Clean input data. Deletes unuseful columns. Correlations columns and outliers are ether deleted or a mask is created.

6.3.2 Function Documentation

6.3.2.1 check_correlated_column()

```
pd.DataFrame scripts.cleaner.check_correlated_column (
    pd.DataFrame df,
    float threshold = 0.9,
    bool remove = False,
    list preserve_columns = None,
    str plot_dir = "plot",
    str data_dir = "data" )
```

Check whether some columns are not correlated over given threshold.
If yes, remove them or create mask.

```
:param df: input DataFrame
:param threshold: correlation threshold
:param remove: if True, remove these columns, otherwise make mask
:param preserve_columns: columns, that are always preserved
:param plot_dir: dir to save generated correlation plot
:param data_dir: dir to store masks
:return: output DataFrame
```

6.3.2.2 check_outliers()

```
def scripts.cleaner.check_outliers (
    pd.DataFrame df,
    float threshold = 4.2,
    bool remove = False )
```

Check whether some outliers are over given threshold (standard deviation).
If yes, remove them, otherwise create mask.

```
:param df: input DataFrame
:param threshold: standard deviation
:param remove: if True, remove these columns, otherwise make mask
:return: output DataFrame
```

6.3.2.3 create_correlation_graphs()

```
list scripts.cleaner.create_correlation_graphs (
    pd.DataFrame correlations,
    str fig_location,
    float threshold )
```

Plots correlation graphs.
:param correlations: correlation values.
:param fig_location: graph image file.
:param threshold: correlation threshold.
:return: list of subgraphs.

6.3.2.4 get_correlated_descriptors()

```
def scripts.cleaner.get_correlated_descriptors (
    pd.DataFrame df,
    int threshold,
    str fig_location )
```

This function creates correlation between descriptors

:param df: input DataFrame
:param threshold: threshold for minimal correlation
:param fig_location: location of correlation figure
:return: created correlations

6.3.2.5 get_duplicat_correlated_descriptors()

```
list scripts.cleaner.get_duplicat_correlated_descriptors (
    nx.MultiGraph graphs,
    exclude )
```

Fnc extract corelated descriptors (column names) from graphs. From each correlation group is excluded one item or if some items from exclude are in correlation, these items are excluded instead.

:param graphs: input graphs
:param exclude: list of descriptors to exclude from final result
:return: list of correlated descriptors without excluded

6.3.2.6 load_data()

```
pd.DataFrame scripts.cleaner.load_data (
    str csv_file_name )
```

Load data from csv ta pandas DataFrame

:param csv_file_name: csv file name
:return: loaded DataFrame

6.3.2.7 main()

```
None scripts.cleaner.main (
    str input_file = "data/chem_output/chem_populated.csv",
    str output_file = "data/final/phototox.csv",
    float correlation_threshold = 0.95,
    float outliers_threshold = 4.2,
    list preserve_columns = None,
    bool remove = False )
```

Runs a series of cleaner functions and writes the result to the output file.

:param input_file: input file location
:param output_file: output file location
:param correlation_threshold: float
:param outliers_threshold: float
:param preserve_columns: list
:param remove: bool
:return: None

6.3.2.8 plot_correlation_heatmap()

```
def scripts.cleaner.plot_correlation_heatmap (
    pd.DataFrame correlations,
    str fig_location,
    int threshold )
```

Plot heatmap of correlated descriptors
:param correlations: input DataFrame with correlations
:param fig_location: path to store plot
:param threshold: used threshold to generate correlations

6.3.2.9 process_config()

```
dict scripts.cleaner.process_config (
    str config_file = "conf/cleaner.ini" )
```

Process input config and extract
:param config_file: config file name location
:return: configuration

6.3.2.10 remove_duplicits()

```
pd.DataFrame scripts.cleaner.remove_duplicits (
    pd.DataFrame df,
    str subset,
    keep = "first" )
```

Removes duplicit rows
:param df: input DataFrame
:param subset: columns where to check duplicity
:param keep: keep argument
:return: output DataFrame

6.3.2.11 remove_useless_columns()

```
pd.DataFrame scripts.cleaner.remove_useless_columns (
    pd.DataFrame df )
```

Remove columns from DataFrame, that are duplicated from multiple sources or otherwise unuseful
:param df: input DataFrame
:return: reduced DataFrame

6.3.2.12 save_mask()

```
None scripts.cleaner.save_mask (
    pd.DataFrame df,
    str mask_name,
    [None, list] columns = None,
    [None, list] rows = None,
    str data_dir = "data" )

Save mask
:param df: data.
:param mask_name: mask string.
:param columns: cols.
:param rows: rows.
:param data_dir: data directory.
:return: None
```

6.3.2.13 set_proper_data_type()

```
pd.DataFrame scripts.cleaner.set_proper_data_type (
    pd.DataFrame df )

Set data types to df columns.
:param df: input DataFrame
:return: df with changed types
```

6.3.2.14 unique_count()

```
[str, int] scripts.cleaner.unique_count (
    pd.Series column )

Get number of unique values in Series
:param column: input Series
:return: columns name and number of unique values
```

6.3.3 Variable Documentation

6.3.3.1 args

```
dict scripts.cleaner.args = process_config(os.path.join(project_dir, "conf/cleaner.ini"))
```

6.3.3.2 input_data

```
scripts.cleaner.input_data = os.path.join(project_dir, "data/chem_output/chem_populated.csv")
```

6.3.3.3 output_data

```
scripts.cleaner.output_data = os.path.join(project_dir, "data/chem_output/phototox.csv")
```

6.3.3.4 project_dir

```
scripts.cleaner.project_dir = os.path.dirname(script_dir)
```

6.3.3.5 script_dir

```
scripts.cleaner.script_dir = os.path.dirname(os.path.realpath(__file__))
```

6.4 scripts.crawlers Namespace Reference

Namespaces

- [abstract_crawler](#)
- [crawler_orchestrator](#)
- [dependencies](#)

6.5 scripts.crawlers.abstract_crawler Namespace Reference

Classes

- class [AbstractCrawler](#)

6.5.1 Detailed Description

Defines an abstract class for crawlers.

6.6 scripts.crawlers.crawler_orchestrator Namespace Reference

Functions

- def [main](#) (file)

6.6.1 Detailed Description

Manages the execution of web crawling.

6.6.2 Function Documentation

6.6.2.1 [main\(\)](#)

```
def scripts.crawlers.crawler_orchestrator.main (
    file )
```

Accesses SwissADME and PubChem to retrieve smile codes and descriptors.
Uses two separate crawlers for each executed in sequence, each producing a separate output.

```
:param file: The input file for the crawler, by default data/input_data/chem.csv
:return None
```

6.7 scripts.crawlers.dependencies Namespace Reference

Namespaces

- [pubchem](#)
- [swissadme](#)
- [swisstarget](#)

6.8 scripts.crawlers.dependencies.pubchem Namespace Reference

Classes

- class [PubchemCrawler](#)

6.8.1 Detailed Description

Crawler of the public chemical database PubChem.

6.9 scripts.crawlers.dependencies.swissadme Namespace Reference

Classes

- class [SwissadmeCrawler](#)

6.9.1 Detailed Description

Crawler of the Swiss Institute of Bioinformatics website for ADME parameters (SwissADME).

6.10 scripts.crawlers.dependencies.swisstarget Namespace Reference

Classes

- class [SwissTargetCrawler](#)

6.10.1 Detailed Description

Crawler of the Swiss Institute of Bioinformatics website for target predictions (SwissTargetPrediction).

6.11 scripts.merger Namespace Reference

Functions

- None [merge \(\)](#)

6.11.1 Detailed Description

Manages the merging function of the data pipeline.

6.11.2 Function Documentation

6.11.2.1 [merge\(\)](#)

None `scripts.merger.merge ()`

Combines the output from PubChem and SwissADME outputs from the crawler, from their default locations.

```
Default input files:  
file_pubchem = 'data/crawler_output/pubchem.csv'  
file_swiss = 'data/crawler_output/swiss.csv'  
  
Default output:  
out = 'data/merger_output/merged.csv'
```

6.12 scripts.populate_chem Namespace Reference

Functions

- pd.DataFrame [populate](#) (file=None)

6.12.1 Detailed Description

Calculating features using rdkit.

6.12.2 Function Documentation

6.12.2.1 [populate\(\)](#)

```
pd.DataFrame scripts.populate_chem.populate (
    file = None )
```

Takes smiles from the output .csv of merger, and populates the csv with chemical descriptors based on them.

The default input path is: "data/merger_output/merged.csv"
the default output path is: "data/chem_output/chem_populated.csv"

The calculated descriptors currently are:

```
cycles
atom_valence
gesteiger sum and mean of positives, negatives, then total mean and total sum
hydrogen_count
csp3
atomic_heavy_atoms
balban_j
hallkier
labute_asa
h_donors and h_acceptors
carbon_count
tpsa
druglikeness
vsa_logP
cycle_type_counts
```

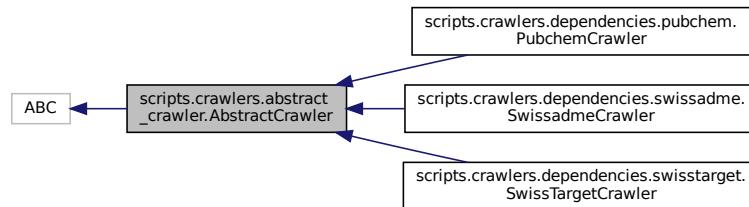
```
:param file: Input file for the function for testing, otherwise leave empty for pipeline automation
:return The modified dataframe, but it's also edited inplace
```

Chapter 7

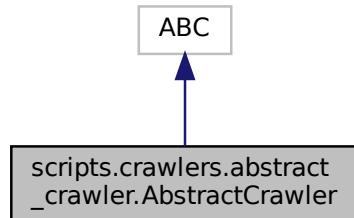
Class Documentation

7.1 scripts.crawlers.abstract_crawler.AbstractCrawler Class Reference

Inheritance diagram for scripts.crawlers.abstract_crawler.AbstractCrawler:



Collaboration diagram for scripts.crawlers.abstract_crawler.AbstractCrawler:



Public Member Functions

- bool `is_file_downloaded` (cls, str filename, int timeout=5)
- None `crawl` (cls, dict prefs)

7.1.1 Detailed Description

Abstract class that is used as an interface for crawler classes.

7.1.2 Member Function Documentation

7.1.2.1 `crawl()`

```
None scripts.crawlers.abstract_crawler.AbstractCrawler.crawl (
    cls,
    dict prefs )
```

Abstract method, has to be implemented. It should contain the core functionality of the crawler.
:param prefs: preferences for the webdriver.
:return: None

Reimplemented in [scripts.crawlers.dependencies.pubchem.PubchemCrawler](#), [scripts.crawlers.dependencies.swisstarget.SwissTarget](#) and [scripts.crawlers.dependencies.swissadme.SwissadmeCrawler](#).

7.1.2.2 `is_file_downloaded()`

```
bool scripts.crawlers.abstract_crawler.AbstractCrawler.is_file_downloaded (
    cls,
    str filename,
    int timeout = 5 )
```

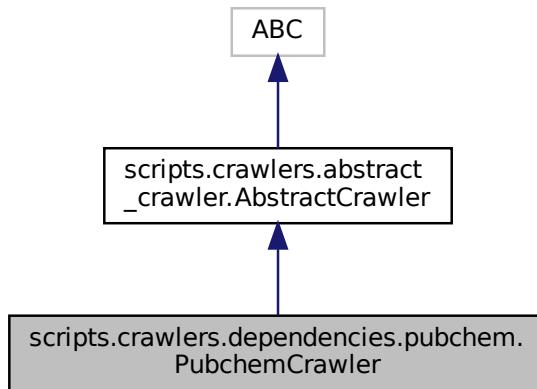
Checks whether the required file is present.
:param filename: path to the file.
:param timeout: time until file search stops.
:return: boolean stating whether the file was found or not.

The documentation for this class was generated from the following file:

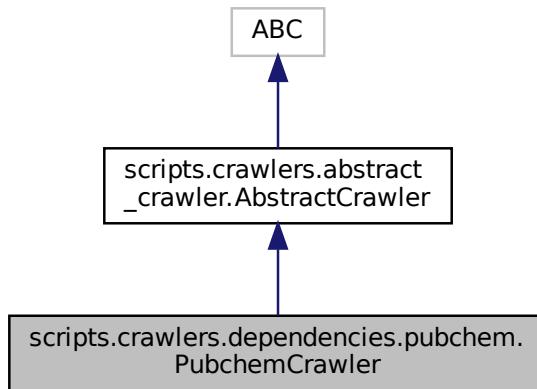
- [scripts/crawlers/abstract_crawler.py](#)

7.2 scripts.crawlers.dependencies.pubchem.PubchemCrawler Class Reference

Inheritance diagram for scripts.crawlers.dependencies.pubchem.PubchemCrawler:



Collaboration diagram for scripts.crawlers.dependencies.pubchem.PubchemCrawler:



Public Member Functions

- str [get_classname](#) (cls)
- None [crawl](#) (cls, dict prefs)
- bool [is_file_downloaded](#) (cls, str filename, int timeout=5)

Static Public Attributes

- list [COLUMNS](#)

7.2.1 Detailed Description

Class for the Pubchem crawler. The class uses Selenium for information extraction.
The crawler uses webdriver for Chrome. The class Follows the :class:`AbstractCrawler` interface.

7.2.2 Member Function Documentation

7.2.2.1 `crawl()`

```
None scripts.crawlers.dependencies.pubchem.PubchemCrawler.crawl (
    cls,
    dict prefs )
```

Extracts relevant information from the Pubchem database. Results are saved into a .csv file.
:param prefs: preferences for the Chrome webdriver.
:return: None

Reimplemented from [scripts.crawlers.abstract_crawler.AbstractCrawler](#).

7.2.2.2 `get_classname()`

```
str scripts.crawlers.dependencies.pubchem.PubchemCrawler.get_classname (
    cls )
```

Get the name of the class.
:return: name of the class as a string.

7.2.2.3 `is_file_downloaded()`

```
bool scripts.crawlers.abstract_crawler.AbstractCrawler.is_file_downloaded (
    cls,
    str filename,
    int timeout = 5 ) [inherited]
```

Checks whether the required file is present.
:param filename: path to the file.
:param timeout: time until file search stops.
:return: boolean stating whether the file was found or not.

7.2.3 Member Data Documentation

7.2.3.1 COLUMNS

```
list scripts.crawlers.dependencies.pubchem.PubchemCrawler.COLUMNS [static]
```

Initial value:

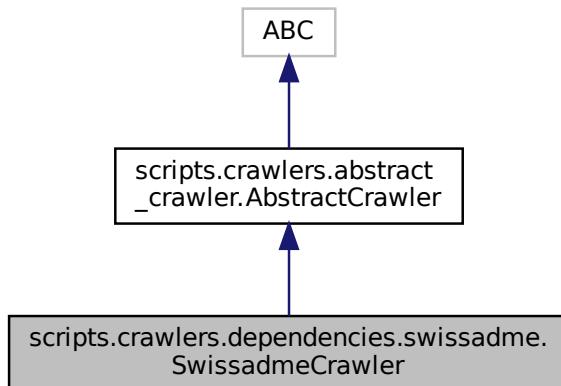
```
= ["Phototoxic", "Name", "Smiles", "Molecular Weight", "Hydrogen Bond Donor Count",
   "Hydrogen Bond Acceptor Count", "Rotatable Bond Count", "Exact Mass",
   "Monoisotopic Mass", "Topological Polar Surface Area", "Heavy Atom Count",
   "Formal Charge", "Complexity", "Isotope Atom Count",
   "Defined Atom Stereocenter Count", "Undefined Atom Stereocenter Count",
   "Defined Bond Stereocenter Count", "Undefined Bond Stereocenter Count",
   "Covalently-Bonded Unit Count", "Compound Is Canonicalized"]
```

The documentation for this class was generated from the following file:

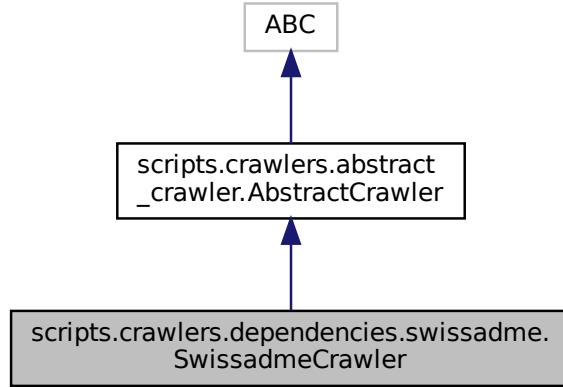
- [scripts/crawlers/dependencies/pubchem.py](#)

7.3 scripts.crawlers.dependencies.swissadme.SwissadmeCrawler Class Reference

Inheritance diagram for scripts.crawlers.dependencies.swissadme.SwissadmeCrawler:



Collaboration diagram for scripts.crawlers.dependencies.swissadme.SwissadmeCrawler:



Public Member Functions

- str [get_classname](#) (cls)
- None [crawl](#) (cls, dict prefs)
- bool [is_file_downloaded](#) (cls, str filename, int timeout=5)

7.3.1 Detailed Description

Class for the SwissADME crawler. The class uses Selenium for information extraction. The crawler uses webdriver for Chrome. The class Follows the :class:`AbstractCrawler` interface.

7.3.2 Member Function Documentation

7.3.2.1 [crawl\(\)](#)

```
None scripts.crawlers.dependencies.swissadme.SwissadmeCrawler.crawl (
    cls,
    dict prefs )
```

Extracts information from the SwissADME website. Results are saved into a .csv file.
:param prefs: preferences for the Chrome webdriver.
:return: None

Reimplemented from [scripts.crawlers.abstract_crawler.AbstractCrawler](#).

7.3.2.2 get_classname()

```
str scripts.crawlers.dependencies.swissadme.SwissadmeCrawler.get_classname (
    cls )  
  
Get the name of the class.  
:return: name of the class as a string.
```

7.3.2.3 is_file_downloaded()

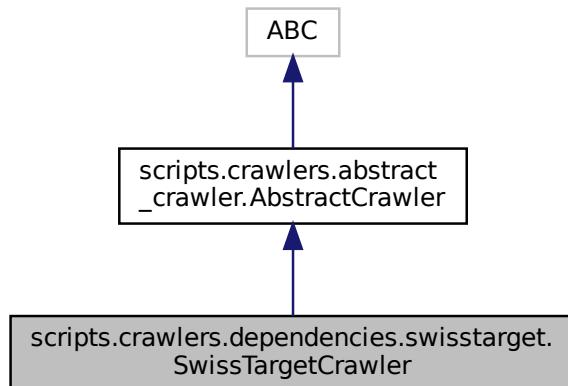
```
bool scripts.crawlers.abstract_crawler.AbstractCrawler.is_file_downloaded (
    cls,
    str filename,
    int timeout = 5 ) [inherited]  
  
Checks whether the required file is present.  
:param filename: path to the file.  
:param timeout: time until file search stops.  
:return: boolean stating whether the file was found or not.
```

The documentation for this class was generated from the following file:

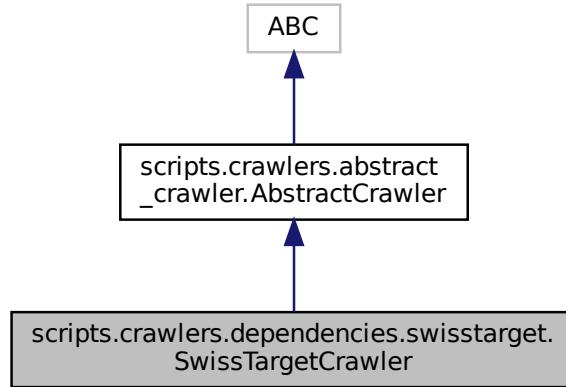
- scripts/crawlers/dependencies/swissadme.py

7.4 scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler Class Reference

Inheritance diagram for scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler:



Collaboration diagram for scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler:



Public Member Functions

- str [get_classname](#) (cls)
- None [crawl](#) (cls, dict prefs)
- bool [is_file_downloaded](#) (cls, str filename, int timeout=5)

Static Public Attributes

- list [COLUMNS](#)

7.4.1 Detailed Description

Class for the SwissTarget crawler. The class uses Selenium for information extraction. The crawler uses webdriver for Chrome. The class Follows the :class:`AbstractCrawler` interface.

7.4.2 Member Function Documentation

7.4.2.1 [crawl\(\)](#)

```
None scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler.crawl (
    cls,
    dict prefs )
```

Extracts information from the SwissTargetPrediction website.
Results are saved into a .csv file.
:param prefs: preferences for the Chrome webdriver.
:return: None

Reimplemented from [scripts.crawlers.abstract_crawler.AbstractCrawler](#).

7.4.2.2 get_classname()

```
str scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler.get_classname (
    cls )
```

Get the name of the class.
:return: name of the class as a string.

7.4.2.3 is_file_downloaded()

```
bool scripts.crawlers.abstract_crawler.AbstractCrawler.is_file_downloaded (
    cls,
    str filename,
    int timeout = 5 ) [inherited]
```

Checks whether the required file is present.
:param filename: path to the file.
:param timeout: time until file search stops.
:return: boolean stating whether the file was found or not.

7.4.3 Member Data Documentation

7.4.3.1 COLUMNS

```
list scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler.COLUMNS [static]
```

Initial value:

```
= ["Compound", "Target", "Common name", "Uniprot ID", "ChEMBL ID",
   "Target Class", "Probability*", "Known actives (3D/2D)"]
```

The documentation for this class was generated from the following file:

- scripts/crawlers/dependencies/[swisstarget.py](#)

Chapter 8

File Documentation

8.1 README.md File Reference

8.2 run_pipeline.py File Reference

Namespaces

- [run_pipeline](#)

Functions

- `def run_pipeline.run_pipe ()`

8.3 scripts/__init__.py File Reference

Namespaces

- [scripts](#)

8.4 scripts/crawlers/__init__.py File Reference

Namespaces

- [scripts.crawlers](#)

8.5 scripts/crawlers/dependencies/__init__.py File Reference

Namespaces

- [scripts.crawlers.dependencies](#)

8.6 scripts/cleaner.py File Reference

Namespaces

- [scripts.cleaner](#)

Functions

- pd.DataFrame [scripts.cleaner.load_data](#) (str csv_file_name)
- pd.DataFrame [scripts.cleaner.set_proper_data_type](#) (pd.DataFrame df)
- [str, int] [scripts.cleaner.unique_count](#) (pd.Series column)
- pd.DataFrame [scripts.cleaner.remove_useless_columns](#) (pd.DataFrame df)
- list [scripts.cleaner.get_duplicat_correlated_descriptors](#) (nx.MultiGraph graphs, exclude)
- def [scripts.cleaner.plot_correlation_heatmap](#) (pd.DataFrame correlations, str fig_location, int threshold)
- def [scripts.cleaner.get_correlated_descriptors](#) (pd.DataFrame df, int threshold, str fig_location)
- list [scripts.cleaner.create_correlation_graphs](#) (pd.DataFrame correlations, str fig_location, float threshold)
- None [scripts.cleaner.save_mask](#) (pd.DataFrame df, str mask_name, [None, list] columns=None, [None, list] rows=None, str data_dir="data")
- pd.DataFrame [scripts.cleaner.check_correlated_column](#) (pd.DataFrame df, float threshold=0.9, bool remove=False, list preserve_columns=None, str plot_dir="plot", str data_dir="data")
- pd.DataFrame [scripts.cleaner.remove_duplicits](#) (pd.DataFrame df, str subset, keep="first")
- def [scripts.cleaner.check_outliers](#) (pd.DataFrame df, float threshold=4.2, bool remove=False)
- None [scripts.cleaner.main](#) (str input_file="data/chem_output/chem_populated.csv", str output_file="data/final/phototox.csv", float correlation_threshold=0.95, float outliers_threshold=4.2, list preserve_columns=None, bool remove=False)
- dict [scripts.cleaner.process_config](#) (str config_file="conf/cleaner.ini")

Variables

- [scripts.cleaner.script_dir](#) = os.path.dirname(os.path.realpath(__file__))
- [scripts.cleaner.project_dir](#) = os.path.dirname(script_dir)
- dict [scripts.cleaner.args](#) = process_config(os.path.join(project_dir, "conf/cleaner.ini"))
- [scripts.cleaner.input_data](#) = os.path.join(project_dir, "data/chem_output/chem_populated.csv")
- [scripts.cleaner.output_data](#) = os.path.join(project_dir, "data/chem_output/phototox.csv")

8.7 scripts/crawlers/abstract_crawler.py File Reference

Classes

- class [scripts.crawlers.abstract_crawler.AbstractCrawler](#)

Namespaces

- [scripts.crawlers.abstract_crawler](#)

8.8 scripts/crawlers/crawler_orchestrator.py File Reference

Namespaces

- [scripts.crawlers.crawler_orchestrator](#)

Functions

- def [scripts.crawlers.crawler_orchestrator.main](#) (file)

8.9 scripts/crawlers/dependencies/pubchem.py File Reference

Classes

- class [scripts.crawlers.dependencies.pubchem.PubchemCrawler](#)

Namespaces

- [scripts.crawlers.dependencies.pubchem](#)

8.10 scripts/crawlers/dependencies/swissadme.py File Reference

Classes

- class [scripts.crawlers.dependencies.swissadme.SwissadmeCrawler](#)

Namespaces

- [scripts.crawlers.dependencies.swissadme](#)

8.11 scripts/crawlers/dependencies/swisstarget.py File Reference

Classes

- class [scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler](#)

Namespaces

- [scripts.crawlers.dependencies.swisstarget](#)

8.12 scripts/merger.py File Reference

Namespaces

- [scripts.merger](#)

Functions

- None [scripts.merger.merge \(\)](#)

8.13 scripts/populate_chem.py File Reference

Namespaces

- [scripts.populate_chem](#)

Functions

- pd.DataFrame [scripts.populate_chem.populate \(file=None\)](#)

Index

args
 scripts.cleaner, 16

check_correlated_column
 scripts.cleaner, 12

check_outliers
 scripts.cleaner, 13

COLUMNS
 scripts.crawlers.dependencies.pubchem.PubchemCrawler,
 25
 scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler,
 29

crawl
 scripts.crawlers.abstract_crawler.AbstractCrawler,
 22
 scripts.crawlers.dependencies.pubchem.PubchemCrawler,
 24
 scripts.crawlers.dependencies.swissadme.SwissadmeCrawler,
 26
 scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler,
 28

create_correlation_graphs
 scripts.cleaner, 13

get_classname
 scripts.crawlers.dependencies.pubchem.PubchemCrawler,
 24
 scripts.crawlers.dependencies.swissadme.SwissadmeCrawler,
 26
 scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler,
 28

get_correlated_descriptors
 scripts.cleaner, 13

get_duplicat_correlated_descriptors
 scripts.cleaner, 14

input_data
 scripts.cleaner, 16

is_file_downloaded
 scripts.crawlers.abstract_crawler.AbstractCrawler,
 22
 scripts.crawlers.dependencies.pubchem.PubchemCrawler,
 24
 scripts.crawlers.dependencies.swissadme.SwissadmeCrawler,
 27
 scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler,
 29

load_data
 scripts.cleaner, 14

main
 scripts.cleaner, 14
 scripts.crawlers.crawler_orchestrator, 18

merge
 scripts.merger, 19

output_data
 scripts.cleaner, 17

populate
 scripts.populate_chem, 20

process_config
 scripts.cleaner, 15

project_dir
 scripts.cleaner, 17

README.md, 31

remove_duplicits
 scripts.cleaner, 15

remove_useless_columns
 scripts.cleaner, 15

run_pipe
 run_pipeline, 11

run_pipeline
 run_pipeline, 11

run_pipe, 11

run_pipeline.py, 31

SOVA mask
 scripts.cleaner, 15

script_dir
 scripts.cleaner, 17

scripts, 11

scripts.cleaner, 12

 args, 16

 check_correlated_column, 12

 check_outliers, 13

 create_correlation_graphs, 13

 get_correlated_descriptors, 13

 get_duplicat_correlated_descriptors, 14

 input_data, 16

 load_data, 14

 main, 14

 output_data, 17

 plot_correlation_heatmap, 14

 process_config, 15

 project_dir, 17

 remove_duplicits, 15

 remove_useless_columns, 15

```
    save_mask, 15
    script_dir, 17
    set_proper_data_type, 16
    unique_count, 16
scripts.crawlers, 17
scripts.crawlers.abstract_crawler, 17
scripts.crawlers.abstract_crawler.AbstractCrawler, 21
    crawl, 22
    is_file_downloaded, 22
scripts.crawlers.crawler_orchestrator, 18
    main, 18
scripts.crawlers.dependencies, 18
scripts.crawlers.dependencies.pubchem, 18
scripts.crawlers.dependencies.pubchem.PubchemCrawler,
    23
    COLUMNS, 25
    crawl, 24
    get_classname, 24
    is_file_downloaded, 24
scripts.crawlers.dependencies.swissadme, 19
scripts.crawlers.dependencies.swissadme.SwissadmeCrawler,
    25
    crawl, 26
    get_classname, 26
    is_file_downloaded, 27
scripts.crawlers.dependencies.swisstarget, 19
scripts.crawlers.dependencies.swisstarget.SwissTargetCrawler,
    27
    COLUMNS, 29
    crawl, 28
    get_classname, 28
    is_file_downloaded, 29
scripts.merger, 19
    merge, 19
scripts.populate_chem, 20
    populate, 20
scripts/__init__.py, 31
scripts/cleaner.py, 32
scripts/crawlers/__init__.py, 31
scripts/crawlers/abstract_crawler.py, 32
scripts/crawlers/crawler_orchestrator.py, 33
scripts/crawlers/dependencies/__init__.py, 31
scripts/crawlers/dependencies/pubchem.py, 33
scripts/crawlers/dependencies/swissadme.py, 33
scripts/crawlers/dependencies/swisstarget.py, 33
scripts/merger.py, 34
scripts/populate_chem.py, 34
set_proper_data_type
    scripts.cleaner, 16

unique_count
    scripts.cleaner, 16
```