**Baigiamojo darbo ataskaita**

## 1. Uždavinys

Sukurti sistemą, kuri remdamasi GNM (German New Medicine) principais, padėtų interpretuoti simptomus pagal vartotojo pateiktą situaciją. Tikslas – ne tik klasifikuoti į konfliktų grupes, bet ir generuoti individualų atsakymą, kuris padėtų vartotojui suprasti galimą pasireiškusio simptomo/ligos psichoemocinę priežastį.

## 2. Nuoroda į GitHub

<https://github.com/lukriste/baigiamasis-clean>

## 3. Savęs įvertinimas

**9/10** – atlikau testavimus, eksperimentavau su modeliais, džiaugiuosi savim ir gautu rezultatu, labiau džiugintų AI tikslūs atsakymai, bet kaip sakoma visada yra kur tobulėti.

## 4. Atsiliepimai apie darbo eigą

* Patiko eksperimentuoti su modelių treniravimu, matyti progreso požymius, laukti kokį atsakymą sugeneruos. Galvoti, kaip patogiau būtų pateikti užklausą ir gauti atsakymą. Dar tikrai ir toliau tobulinsiu kaip įrankį. Žadu pridėti kitokių duomenų iš kitos psichosomatikos krypties, kad būtų daugiau ir kuo masiškiau pritaikoma.
* Nepatiko ilgai trunkantis testavimas, supratau kad kompiuteris tam visai netinkamas, buvo sudėtinga suderinti TF/Keras ir Python 3.12.

## 5. Duomenų aprašymas

Duomenis gavau iš knygos, sudėjau į Excel failą kuriame pateikti simptomai, konfliktinės situacijos (ivestis) ir GNM pagrįsti atsakymai. Duomenis perkėliau į MySQL naudojant SQLAlchemy. Prieš importavimą atlikau tekstų normalizavimą: pašalinau nereikalingus simbolius, tvarkiau tarpus, specialiuosius simbolius.

Pirminė GNM konflikto reikšmė (Y) buvo unikali kiekviename įraše, todėl dariau reikšmių grupavimą į konfliktų grupes (pvz., „savigarbos sumažėjimo konfliktas“, „jutiminis konfliktas“ ir pan.), kad būtų galima taikyti klasifikacinius metodus. X reikšmė – žmogaus aprašyta pateiktas simptomas ar aprašyta situacija (pvz. „skauda gerklę“). Bet supratau, kad man netiko tokie atsakymai tik klasifikuota grupės išvestis, norėjau individualių atsakymų.

Vėliau duomenis transformavau modelio treniravimui:

* X: apjungtas tekstas – „simptomas: [simptomas]. [ivestis]“
* Y: modelio generuojama interpretacija – GNM paaiškinimas (target)

Modelių, kuriuos pasirinkau aprašymas

Darbo metu išbandžiau keli skirtingus modelius:

* **TF-IDF + Naive Bayes** – tai klasifikavimo modelis, kuris tekstus vektorizuoja pagal žodžių dažnumą. Naive Bayes daro prielaidą, kad visi žodžiai yra nepriklausomi. Paprastas, bet greitas metodas, tinkamas trumpiems tekstams.
* **TF-IDF + Random Forest** –susideda iš daugelio sprendimo medžių, kurie kartu priima sprendimus. Šiek tiek tikslesnis už Naive Bayes, bet nelabai tinkamas interpretacijos uždaviniams.
* **T5-small** – tai Transformer tipo neuroninis tinklas, konkrečiai: **Seq2Seq (sequence-to-sequence)** architektūra. Šie modeliai naudoja encoder-decoder sistemą, kur pirmiausia tekstas užkoduojamas į skaitmeninę formą, o tada dekoduojamas į atsakymą.
* **mt5-small** – tai daugiakalbė versija T5, taip pat Transformer architektūros. Šis modelis moka daugiau kalbų, tačiau ne visada gerai susitvarko su lietuvių kalba.
* **t5-base iš LukasStankevicius/t5-base-lithuanian-news-summaries-175** – tai jau **specialiai lietuvių kalbai pritaikytas** Transformer modelis. Pasirodė prasmingiausias, generavo sakinius su geresne struktūra, net jei ne visada tiksliai. Jis taip pat yra **Seq2Seq** tipo neuroninis tinklas, pagrįstas Attention mechanizmais, kurie leidžia geriau suprasti kontekstą.

Visi naudoti LLM modeliai (T5, mt5) yra **Feedforward** tipo neuroninių tinklų pagrindu, bet jie naudoja sudėtingą dėmesio (attention) architektūrą, kuri leidžia „skaityti visą tekstą vienu metu“.

RNN ir CNN šiuo atveju nenaudojau – netiko mano duomenims ir norams.

## 6. Darbo uždaviniai

* Surinkti duomenis iš knygos/ žinyno, išversti į lietuvių kalbą, pritaikyti naudojimui ir sudėti į Excel
* Įkelti Excel duomenis į MySQL bazę, apdoroti tekstus ir normalizuoti struktūrą.
* Sukurti Flask aplikaciją, leidžiančią vartotojui įvesti simptomus ir gauti interpretaciją.
* Išbandyti ir įgyvendinti klasifikacinius modelius su TF-IDF + Naive Bayes bei TF-IDF + Random Forest, įvertinti jų tikslumą.
* Generuoti konfliktų interpretacijas naudojant LLM modelį (T5-small, vėliau T5-base, lietuviškai).
* Sukurti sistemą, leidžiančią automatiškai vertinti atsakymų kokybę (BLEU, ROUGE).
* Vizualizuoti treniravimo eigą ir modelio progresą (loss kreivė).

## 7. Rezultatų lentelė

*(Lentelė bus pildoma vėliau. Pradžia pridėta kaip pavyzdys)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bandymas** | **Modelis** | **Parametrai** | **Tikslumas** | **BLEU** | **ROUGE** | **Pastabos** |
| 1 | TF-IDF + Naive Bayes | StopWords=No, ngram=1 | 0.25 | - | - | Daug klasių, prastas generalizavimas |
| 2 | TF-IDF + Naive Bayes | StopWords=Yes | 0.23 | - | - | Dar blogesnis rezultatas |
| 3 | TF-IDF + Random Forest | 100 estimators | 0.29 | - | - | Pagerėjo, bet vis tiek per mažai |
| 4 | T5-small | 3 epoch, max\_len=128 | - | 0.07 | 0.24 | Pasikartojimai, prasti sakiniai |
| 5 | gnm-t5-liet-var2 | 20 epoch, max\_len=128, bs=4 | - | 0.08 | 0.28 | Prasmingesni sakiniai |

*(Lentelė bus papildyta iki 30+ eilučių galutiniame variante)*

## 8. Pastebėjimai treniravimo metu

Pradžioje duomenis buvo sudėti į MySQL duomenų bazę, tačiau vėliau perkėliau į Railway dėl lankstumo ir patogesnio naudojimo su aplikacijomis, pagalvojau kad ateitį bus patogiau.

Pirmiausia išbandžiau prižiūrimojo mokymosi klasifikatorius – Naive Bayes ir Random Forest. Naudotas TF-IDF, tačiau:

* Naive Bayes duoda greitus, bet nekokybiškus rezultatus, ypač kai klasės nesubalansuotos.
* Random Forest šiek tiek pagerino rezultatus, bet vis tiek neatitiko tikslų – norėjosi interpretacijos, ne tik klasės.

LLM bandymų pradžioje naudojau google/mt5-small ir t5-small modelius Google Colab aplinkoje, tačiau gauti prasti rezultatai, dažnai pasikartojimai, nesugeneruoti prasmingi sakiniai. Bandžiau pridėti repetition\_penalty, nes buvo gana daug pasikartojimu, tačiau nepadėjo.

Pastebėjau, kad t5-small modelis geriau generuoja sakinius, nors be lietuviškų simbolių. Vėliau perėjau prie specializuoto modelio LukasStankevicius/t5-base-lithuanian-news-summaries-175, kurį treniravau jau lokaliai su TensorFlow. Modelis nebuvo perkeltas su svoriais – viskas buvo mokoma iš naujo, todėl negalėjau daug eksperimentuoti su hiperparametrais dėl ilgo treniravimo proceso.

Geriausi rezultatai gauti su gnm-t5-liet-2var modeliu, kuris treniruotas 20 epochų su 240 įrašų. Loss kreivės rodo stabilų mažėjimą. Generuoti sakiniai tapo prasmingesni, tačiau dar ne idealiai tikslūs. Visgi modelio kryptis aiški ir naudinga realiam žmogui.

## 9. Išvados

* Pavyko pasiekti pagrindinį tikslą – turėti veikiančią sistemą, kuri interpretuoja simptomus
* Klasifikaciniai modeliai buvo riboti, o LLM generacija parodė daugiau potencialo
* Nepavyko išgauti labai aukštos BLEU ar ROUGE vertės, tačiau pavyko pasiekti, kad atsakymai taptų prasmingi, kol kas nelabai naudingi žmogui
* Treniravimo progresas matosi grafike T5\_treniravimo grafikas, geriausias modelis: gnm-t5-liet-2var

## 10. Papildoma

* Į sistemą įdiegiau vertinimo metriką, kur automatiškai po kiekvienos užklausos ROUGE ir BLEU įrašomi į CSV
* Vartotojas gali įvesti simptomą tiek iš sąrašo, tiek individualiai, nebeužteko laiko patobulinti sąraše esančius simptomus, kad būtų paprastesni
* Sukurta paprasta sąsaja su duomenų baze, kur kaupiami atsakymai (įrašai klasifikuojami kaip iš „excel“ arba „vartotojo“ šaltinio ir vertinimas modelio atsakymo, ar teigiamas ar neigiamas)
* Planuoju toliau plėsti bazę duomenų, pateikti kad vartotojai galėtų naudotis ir tuo pačiu man pildyti bazę, kad po kiek laiko ją galėčiau vėl pertreniruoti ir papildyti, kad su laiku AI atsakymai būtų kuo tikslesni.