Zadanie 2 – Softverove spracovanie udajov a procesov

Program machine\_learning

Uloha 1: Pridajte do kódu ďalší model strojového učenia (ľubovoľný), a taktiež definujte parametre a ich hodnoty pre Grid Search.

Pre tento typ ulohy o rozhodovani benignych a malignych buniek som sa rozhodol pre model Random Forest, pretoze ide o robustny algoritmus vhodny na binarne klasifikacne ulohy. Tento model kombinuje viacero rozhodovacich stromov, cim znizuje riziko preucenia a dosahuje vyssiu stabilitu vysledkov.

Obrázok, na ktorom je text, snímka obrazovky

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Jednotlive parametre predstavujú:

* 'n\_estimators' - pocet stromov, cim vysie cislo lepsia presnost, ale dlsi vypocet
* 'max\_depth' - maximalna hlbka stromov, (None) znamena, ze kazdy strom sa rozvetvi uplne
* 'min\_samples\_split' a 'min\_samples\_leaf' - minimalny pocet vzoriek pri deleni a vo vetve

Uloha 2: Implementujte ďalšiu (ľubovoľnú) metriku pre evaluáciu modelov

Do evaluacie modelu som pridal metriku **Recall**, pretoze je dolezite minimalizovat pocet falosne negativnych pripadov, teda situacii, ked model nespravne oznaci malignu bunku ako benignu. Vyjadruje, aky podiel skutocne malignych pripadov model spravne identifikoval. Napriklad hodnota rec = 0.9258 znamena ze model spravne zachytil 92.58% vsetkych skutocne malignych buniek. Ak je nizka hodnota (recall), tak model prehliada maligne bunky, co je nespravne a v tomto pripade rizikove.

Obrázok, na ktorom je text, písmo, snímka obrazovky

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Ukladanie do logov v subore model\_accuracies.csv

Obrázok, na ktorom je text, snímka obrazovky, písmo, informácia

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Pridanie do grafov pre funkciu hustoty rozdelenia

Obrázok, na ktorom je text, snímka obrazovky, vývoj, diagram

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Graf zobrazujuci priebeh metriky Recall pocas replikacii

Obrázok, na ktorom je text, rad, diagram, vývoj

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Uloha 3: Do implementácie pridajte ukladanie všetkých grafov, ktoré sa vytvárajú pri behu skriptu main.py`` v adresári machine\_learning`

Do suboru s nazvom plotting/ base\_plotter.py konkretne do metody \_generic\_plot som pridal logiku na ukladanie vsetkych grafov do priecinku /outputs/ pomocou plt.savefig()

Obrázok, na ktorom je text, písmo, snímka obrazovky

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Uloha 4: V skripte main.pynastavte počet replikácií na vyššie číslo (rozumne, podľa vlastného uváženia). Vykonajte beh aplikácie s Vašou implementáciou. Po skončení behu zanalyzujte vygenerované grafy a pár vetami popíšte ich interpretáciu. (Napr. v čom je ktorý ML model lepší, a pod.)

Rozhodol som sa pre pocet replikacii = 30.

Obrázok, na ktorom je text, snímka obrazovky, písmo, softvér

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Ako prve beh programu trval podstatne dlhsie (5 min), pretoze pocet replikacii bol trojnasobny. Na zaklade vykreslenych grafov mozno povedat, ze pre vsetky grafy pre fukciu hustoty rozdelenia je model Logistic Regression lepsi ako model Random Forest (vrchol je blizsie ku 1). Grafy priebehu replikacii taktiez naznacuju ze model Logistic Regression vykazuje lepsie hodnoty ako moj sledovany model strojoveho ucenia. Nie uplne by som suhlasil s vyrokom ze s narastajucom poctom replikacii sa model lepsie prisposobi datam alebo ze je stabilnejsi. Jedna taka vec co som si vsimol, tak model Random Forest ma na zaciatku nizsie hodnoty, ktore sa neskor zlepsia. Myslim si ze model Logistic Regression je celkom slusny na tento typ problemu.

Obrázok, na ktorom je text, rad, diagram, vývoj

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Obrázok, na ktorom je text, rad, diagram, vývoj

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.