DataVaders - Projektterv

Bajnóczi Bendegúz, Csiszár András, Mészáros Zsolt, Szakál Mátyás, Vass Máté
Gépi tanulási módszerek gyakorlat – 2022

Formai követelmények

Minden szekció maximum **1** oldal hosszú lehet. Betűméret: **12**. Ahol lehet, folyó szövegben fogalmazzunk listák és táblázatok helyett, hacsak nem tudunk azokkal sok helyet megspórolni. **Maximális pontszám** eléréséhez az összes felsorolt kérdésre válaszolni kell, valamint saját ötleteket is tartalmaznia kell a leírásnak, ezzel bizonyítva, hogy a csapat mélyen megvizsgálta a feladatot.

Dokumentáció beadása

Az elvárt formátum: pdf, ezt CooSpace-en kell beadni. Minden beadás tartalmazza a korábbi szekciókat is, például a 2. Mérföldkő minden előtte lévő Mérföldkövet és szekciót tartalmaz.

Mérföldkő neve	Pontszám	Mérföldkő határideje	Dokumentáció beadásának határideje
Feladatleírás	+5	-	2022-02-28
Adatfeldolgozás (MK - I)	20	2022-03-07	2022-02-28
Modellezés (MK - II)	20	2022-03-28	2022-03-21
További fejlesztések (MK - III)	20	2022-05-02	2022-04-25
Projekt prezentálása	5	2022-05-09	-
Egyéni feladatok	35	2022-05-02	2021-04-25

Feladat leírása

- Mi a feladat?
    - A projekt során egy felügyelt gépi tanulási problémát oldunk meg egy általunk tetszőlegesen választott adathalmazon. A munkához egy banki adatbázist választottunk, amiben a bank az ügyfeleiről tárol el különféle adatokat. Ezek alapján azt szeretnénk megjósolni, hogy egy új ügyfél a jövőben várhatóan fog-e betéti számlát nyitni.
- Feladat típusa
    - Osztályozás.
- Motiváció
    - A meglévő adatok alapján megpróbálhatjuk megjósolni, hogy egy új ügyfél tervez-e megtakarítást elhelyezni a bankban.
    - Miért hasznos ez a feladat?
        i. Így céltudatosan tud a bank hirdetéseket és telefonos ajánlatokat tenni bizonyos ügyfeleknek.
- Adathalmaz leírása
    - Adat formátuma
        i. .csv
    - Adatsorok jellemzői
        i. Rendezett adatsoraink vannak
        ii. Jellemzők:
            1. *Kliens adatai*: életkor, munka fajtája, kapcsolati státusz, tanulmányok, van-e hiteltartozása, átlagos éves egyenleg, van-e lakáshitele, van-e személyi hitele
            2. *Legutóbbi kapcsolatfelvétel jellemzői:* kapcsolatfelvétel típusa, kapcsolatfelvétel hónapja, kapcsolatfelvétel napja, kapcsolatfelvétel időtartama
            3. *Egyéb jellemzők:* kapcsolatfelvételek száma, utolsó kapcsolatfelvétel óta eltelt napok száma, eddigi összes kapcsolatfelvétel száma, előző kapcsolatfelvétel kimenetele
        iii. Címke:
            1. Az ügyfélnek van-e a banknál betéti számlája?
    - Prediktálási cél
        i. F-e az ügyfél a jövőben betéti számlát nyitni?
    - Adathalmaz mérete
        i. 45211 felcímkézett rekord
        ii. 16 jellemző vektor
        iii. 1 bináris (igen/nem) címke
- Használt környezet és eszközök:
    - Programozási nyelv
        i. Python (Google Colaboratory Notebook)
    - Gépi tanulást megvalósító könyvtárak
        i. NumPy, pandas, scikit-learn
    - Verziókövetés
        i. Github repository: https://github.com/material1999/gepitan_project

Adatfeldolgozás (MK - I)

- Adatfelosztási módszer
    - Mivel nagyon nagy mennyiségű felcímkézett példa áll rendelkezésünkre, így terveink szerint 70%-15%-15% felosztást fogunk alkalmazni (70% train, 15% dev, 15% test)
    - Így nagyjából 30 ezer adaton tudunk majd betanítani, továbbá 7-7 ezer adaton fejleszteni, valamint tesztelni.
- Tanuláshoz felhasznált jellemzők
    - A tanuláshoz terveink szerint az adathalmazunk jellemzésénél felsoroltak közül az *“1. Kliensek jellemzői”* attribútumokat szeretnénk felhasználni. Ezeket gondoltuk a legfontosabb információknak a feladatunk szempontjából, a többi jellemző kihagyásával elkerülhetjük az esetleges túltanulást.
    - Összehasonlításképp tervezzük a modellünk tanítását egy bővített jellemző halmazon is elvégezni, és az így kapott eredményeinket különböző metrikák szerint összehasonlítani a szűkebb halmazon kapottakkal.
- Mi fog történni az üres cellákat tartalmazó rekordokkal, ha vannak?
    - Mivel azoknak a rekordoknak a száma, amikben van üres cella, nem jelentős, ezért egyszerűen el fogjuk dobni őket (körülbelül 2 ezer rekord a 45 ezerből). Azért is döntöttünk így, mert 1-1 hiányzó adatot nem egyértelmű, mely másik rekordok alapján lehet hatékonyan prediktálni.
- Előfeldolgozási lépések
    - A tanításhoz használt jellemző vektorok közül a szöveges információk csak adott, igen kis méretű halmazokból vehetnek fel értékeket. Így ezeket először egy OrdinalEncoder segítségével egyenletes eloszlású számsorrá alakíthatjuk, majd egy MinMaxScaler segítségével lenormálhatjuk a [0,1] intervallumba.
    - Ezek után a többi, számszerű adatot tartalmazó jellemző vektort egy StandardScaler segítségével fogjuk transzformálni, ezzel a kapott oszlopokban az átlagunk 0, a szórásunk pedig 1 lesz.
    - A két transzformáció eredményét összesítve készen is áll az adathalmazunk a később definiált modellünk betanítására.
- A futtatásokhoz jelenleg nem tervezünk külső szkript-eket írni, a futtatási paramétereket külön kód blokkokba szervezzük ki a egyes szekciók elején, és minden programkód a Google Colaboratory Notebook-ban lesz megtalálható.

Modellezés (MK - II)

- Az értékeléshez használt metrikák
    - Mivel a feladatunk osztályozás, ezért a scikit-learn “Metrics and scoring” oldalának tanulmányozásával (link) az alábbi metrikákat választottuk:
        i. **accuracy_score** (A többcímkés osztályozás esetén a függvény visszaadja a részhalmaz pontosságát.)
        ii. **balanced_accuracy_score** (mivel nem egyenletesen oszlanak el a címkéink, ezért a pontosságot ezek súlyozásával is megkaphatjuk.)
        iii. **recall_score** (Egy arányt ad vissza. (TP / (TP + FN)) , ahol TP a true-positive, FN pedig a fals pozitív eredmények száma. Segítségével meghatározhatjuk az összes pozitív mintát.)
        iv. **precision_score** (Egy arányt ad vissza. (TP / (TP + FP)), ahol TP a true-positive, FN pedig a fals negatív. Segítségével elkerülhetjük, hogy a negatív címkék pozitívnak legyenek kategorizálva.)
        v. **f1_score** (A recall_score és a precision_score-nak a harmónikus középértéke. A harmonikus középérték miatt ez az érték csak akkor lesz magas, ha mindkettő metrika magas.)
        vi. **brier_score_loss** (A Brier-pontszám az előrejelzett valószínűség és a tényleges eredmény közötti átlagos négyzetes különbséget méri.)
        vii. **confidence interval** (Valószínűségi intervallum. Azt adja meg, hogy valószínűleg minden érték tartományba fog esni az eredményünk. A konfidencia intervallum adott szignifikancia-szinten: a becsült változó alsó és felső korlátja.)
- Baseline metódus
    - Először egy **véletlenszerű felcímkézést** végzünk el a teszt halmaz elemein. Ekkor nem az 50%-os pontosság az elvárt, hiszen sokkal több “no” címkénk van az adatbázisunkban, mint “yes”.
    - Ehelyett csinálhatunk a címkék arányának ismeretében egy **adott eloszlású véletlen címkézést**, azaz pl. a dobott címkék 80%-a “no” lesz, 20%-a “yes”, ezek pedig véletlenszerűen lesznek elosztva.
    - Ezek után kipróbáljuk a **leggyakoribb címkével** való predikciót.
- Ötletek a modellekhez
    - **Gaussian Naive-Bayes** (A Bayes-tételt alapján az egyes változók között feltételes függetlenséget feltételezve, normál eloszlást felhasználva számolja ki az ismeretlen példányhoz tartozó legvalószínűbb címkét.)
    - **Gaussian Mixture Model** (A Gauss-keverék modell egy valószínűségi modell, amely feltételezi, hogy az összes adat pontot véges számú, ismeretlen paraméterű Gauss-eloszlás keverékéből állítják elő.)
    - **K Nearest Neighbour** (Adott számú legközelebbi szomszéd távolságával súlyozva állapítja meg az ismeretlen példány címkéjét.)
    - **Nearest Centroid** (A Nearest Centroid osztályozó egy egyszerű algoritmus, amely minden osztályt a tagok súlypontjával reprezentál.)
- Optilamizálandó hiperparaméterek
    - **GMM** esetén különböző függőségek kipróbálása
    - **KNN** esetében “legjobb” k megtalálása → Grid Search (Kipróbálja az általunk megadott lehetőségeket, és a legjobb megoldást választja.)