Python programozás adatelemzéshez

Házi feladat

2025 tavasz

A feladatok megoldásához az előadásokon is megismert pandas, numpy, matplotlib és seaborn könyvtárakra lesz szüksége. Az összes feladatot egy notebookban (.ipynb) oldja meg. Ahol szükséges ott a szöveges válaszokat, magyarázatokat markdown cellákba írja. A második részletben egyértelműen jelezze, hogy melyik feladathoz melyik ábra vagy magyarázat tartozik.

A fájlt, amiben a megoldásai vannak úgy adja be, hogy az abban lévő cellák egymás követően hibamentesen lefussanak és a megoldásai láthatóak legyenek. A fájl neve ”<neptun kód>_python_adat” legyen.

Feladatok

1. Az első részletben egy saját függvényt kell készítenie, aminek a segítségével egy egyedi számot fog létrehozni a neve és a neptun kódja alapján.

   1.1. A függvény 3 bemeneti paraméterrel rendelkezzen: a hallgató neve (str), a hallgató neptun kódja (str) és egy szám (int), ami alapméretezettként meg van adva a függvényben. Utóbbi, bármilyen tetszőleges szám lehet.

   1.2. Legyen egy `if/else` elágazás a függvényben, amivel ellenőrzi, hogy a megadott neptun kód megfelelő karakterszámú. Ha nem megfelelő akkor a függvény a ”Helytelen neptun kód” szöveget adja vissza.

   1.3. Hozzon létre 1-1 listát a név és neptun kód karaktereiből úgy, hogy a listákban minden egyes karakter a lista 1-1 eleme. Például, ha a neve Teszt Elek, akkor a lista ['T', 'e', 's', 'z', 't', ' ', 'E', 'l', 'e', 'k'] legyen.

   1.4. A névhez tartozó listában törölje a 3. elemet és a neptun kódhoz tartozóban pedig a 4. elemet cserélje le a neve utolsó betűjére.

   1.5. Az előbb létrehozott két lista elemeit adja be a beépített `ord()` függvénynek és tárolja az eredményeket két új listában. Az `ord()` minden egyes karaktert egy számmá fog alakítani.

   1.6. Hozzon létre egy dictionary-t amiben a key value-k a név és a neptun kód karakterei és az ezekhez társított értékek pedig az `ord()` függvényből kapott értékek legyenek.

   1.7. Hozzon létre egy `for` ciklust, amiben az előbb létrehozott dictionary minden páros indexű elemének az értékéhez hozzáadja az alapméretezett bemeneti értéket, a páratlan indexűekből pedig kivonja azt.

   1.8. Adja össze az összes értéket, ami a dictionary-ben található és tárolja egy változóban. Ez a szám legyen a függvény kimenete. Emellett a függvény jelenítse meg az alábbi szöveget helyesen kitöltve: "Név: , Neptun kód: , Generált szám: ".

   1.9. Végül futtassa a létrehozott függvényt és mentse a kimenetét egy változóba.

2. A második részletben a Stroke Prediction adatszett egy részletét fogja vizsgálni. Ez az adathalmaz arra szolgál, hogy előre jelezze, egy beteg hajlamos-e szélütésre a megadott paraméterek, például nem, életkor, különféle betegségek és dohányzási szokások alapján. Az adatok minden egyes sorában egy beteg jellemzői találhatók. A feladatok során minden ábránál megfelelően legyenek feliratozva a tengelyek és azok jól láthatóak legyenek. Ahol szükséges ott a tengelyek skálázását is változtassa. Az adathalmaz eredeti verziója publikusan elérhető:
   - Kaggle - https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

   2.1. Olvassa be a hf_data_vegyesz.csv-t, az index az *id* oszlop legyen. A `.sample()` segítségével mintavételezzen 2000 adatpontot, úgy, hogy a `.sample()` random_state paraméterének az első feladatban generált számot adja meg. Mire szolgál a random_state paraméter?

   2.2. A teljes adathalmazból válassza ki azokat az adatpontokat, amik nem kerültek be a 2000 mintavételezett közé. Ezek közül az első 20 sort illessze hozzá a mintavételezett adatpontokhoz. A további feladatokat az így létrejött 2020 adatpontból álló adathalmazon végezze el.

   2.3. Hány sorból és oszlopból áll az adathalmaz? Milyen típusú oszlopok vannak és melyik típusból hány darab van?

      2.3.1. Valamennyi oszlophoz írassa ki, hogy hány darab egyedi érték van benne.

      2.3.2. Válasszon ki 2 olyan oszlopot, amit érdemes lenne kategorikus változóként kezelni és végezze is el az átalakítást. Indokolja a választásait!

   2.4. Összesen hány darab hiányzó érték (nan) van az adathalmazban? Melyik oszlopban van a legtöbb hiányzó adat? Ábrázolja oszlopdiagramon, hogy azokban az oszlopokban ahol van legalább 1 nan érték hány darab hiányzó adat van összesen. Az ábrán csökkenő sorrendben legyenek az értékek.

   2.5. Az a*ge* oszlop egyes értékeiből hány darab van? Ezeket ábrázolja hisztogramon.

      2.5.1. Továbbá, készítsen egy olyan több hisztogramból álló ábrát, amin az látható, hogy az egyes munkakategóriák (*work_type*) esetén hogyan alakul az életkor (*age*) eloszlása a stroke-ot kapott betegek körében.

   2.6. Csoportosítsa az adatokat az *age* értékek szerint és ezek alapján számolja ki az átlag és medián értékeit az avg_glucose_level oszlopnak. Ábrázolja az átlagok és mediánok eloszlását egy közös ábrán.

      2.6.1. Pótolja ki a hiányzó értékeket az avg_glucose_level oszlopában, úgy, hogy a pótlás mindig az adott életkorhoz (*age*) tartozó átlagos avg_glucose_level értékkel történjen. Ábrázolja újra az átlagok és mediánok eloszlását egy közös ábrán.

      Az eloszlások ábrázolásánál mindegyik esetben az értékek kernel sűrűségbecslése (KDE) is legyen látható.

   2.7. Készítsen 1-1 kördiagramot a *smoking_status* egyedi értékeihez, amik a stroke-os és nem stroke-os esetek arányát ábrázolják. A *smoking_status* nan értékéhez ne jelenítsen meg diagramot. Az egyes körcikkelyeken jelenítse meg az ahhoz tartozó százalékos értéket is.

   2.8. A `pairplot()` függvénnyel ábrázolja az *age*, az *avg_glucose_level* és a *bmi* paraméterek közötti összefüggéseket és színezéshez használja a *hypertension* változót.

2.8.1. A válasszon ki egy változó párost a fenti három közül. Majd válasszon egy olyan vizualizációs függvényt, aminél a szórásdiagram mellett az értékek eloszlásai is látszanak. Az adatpontokat ebben az esetben is színezze a *hypertension* változó szerint.

2.8.2. Az `lmplot()` függvénnyel készítsen egy összetett ábrát, amin *hypertension* kategóriánként van illesztve 1-1 regressziós egyenes az értékekre.

2.9. Ábrázolja a *smoking_status* oszlop egyedi értékeihez tartozó adatpontok *age* értékeit 1-1 hegedűdiagrammal egy ábrán. Mi figyelhető meg az ábrán?

2.10. Készítsen egy ábrát, amin az *avg_glucose_level* látható a *bmi* függvényében, azoknál az adatpontoknál, ahol a *páciens* nő nemű és életkora kisebb mint a medián. Az adatpontokat az *age* oszlop értékei alapján színezze. Utóbbihoz készítsen egy folytonos színskálát (colorbar). Mi figyelhető meg az ábrán?

2.11. A `pivot_table()` segítségével hozzon létre egy olyan táblázatot amiben az oszlopokat a *work_type* oszlop egyedi értékei adják, a sorok pedig az 'age', az 'avg_glucose_level' és a 'bmi' oszlopok átlag értékeit tartalmazzák az adott work_type-ok esetén. Egy 5 oszlopból és 3 sorból álló táblázatot várunk.

2.11.1. Hozzon létre egy új oszlopot ami a meglévő oszlopok értékeinek az átlagát tartalmazza minden egyes sorban.

2.11.2. A létrehozott táblázatot rendezze csökkenő sorrendben a létrehozott átlag oszlop szerint és mentse el egy excel fájlba(.xlsx), amiben a munkafüzet neve a neptun kódja.

Oszlopok jelentése

id: unique identifier
gender: "Male", "Female" or "Other"
age: age of the patient
hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
ever_married: "No" or "Yes"
work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
Residence_type: "Rural" or "Urban"
avg_glucose_level: average glucose level in blood
bmi: body mass index
smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
stroke: 1 if the patient had a stroke or 0 if not